AGARD-AR-343

AGARD-AR-343

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT
7 RUE ANCELLE, 92200 NEUILLY-SUR-SEINE, FRANCE

AGARD ADVISORY REPORT 343

Integrated Vehicle Management Systems

(Systèmes de gestion de vecteur intégré)

This report has been prepared as a summary of the deliberations of Working Group 1 of the Mission Systems Panel of AGARD.

NORTH ATLANTIC TREATY ORGANIZATION

Published April 1996

Distribution and Availability on Back Cover

AGARD-AR-343

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT
7 RUE ANCELLE, 92200 NEUILLY-SUR-SEINE, FRANCE

AGARD ADVISORY REPORT 343

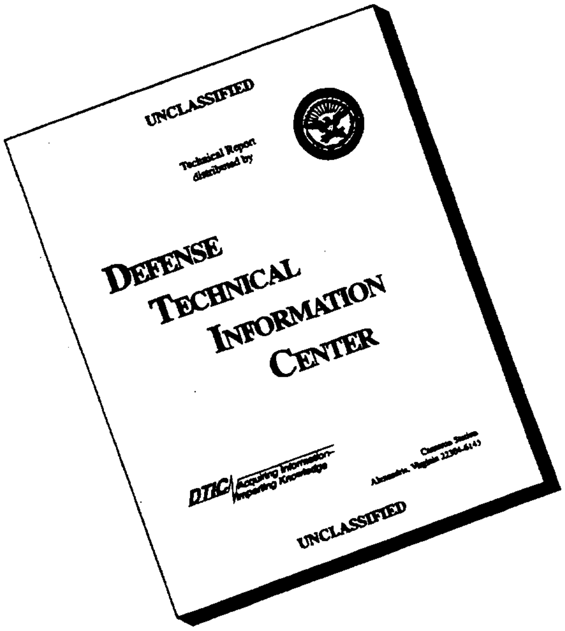
Integrated Vehicle Management Systems

(Systèmes de gestion de vecteur intégré)

This report has been prepared as a summary of the deliberations of Working Group 1 of the Mission Systems Panel of AGARD.

North Atlantic Treaty Organization
Organisation du Traité de l'Atlantique Nord

DISCLAIMER NOTICE



UNCLASSIFIED

Technical Report
distributed by

DEFENSE
TECHNICAL
INFORMATION
CENTER

DTIC Acquiring Information—
Imparting Knowledge

Cameron Station
Alexandria, Virginia 22304-6145

UNCLASSIFIED

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

The Mission of AGARD

According to its Charter, the mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community;
- Providing scientific and technical advice and assistance to the Military Committee in the field of aerospace research and development (with particular regard to its military application);
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Exchange of scientific and technical information;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced
directly from material supplied by AGARD or the authors.

Published April 1996

ISBN 92-836-1035-0

Printed by Canada Communication Group
45 Sacré-Cœur Blvd., Hull (Québec), Canada K1A 0S7

Integrated Vehicle Management Systems

(AGARD AR-343)

Executive Summary

Major trends in technology, weapon system performance goals and least cost aerospace systems are all topical questions. Within electronic systems such concepts can be simplified in the form of the essential following major headings:

- Weapon System Performance;
- Technology;
- Least Cost.

This report looks at one essential integration of the above selected trends: an integration of technology to provide high performance electronic systems at affordable prices. Many questions can be answered in careful, coherent design and development. The desire to functionally integrate subsystems so as to achieve better performance has been greatly assisted by discoveries in the field of data processing. In the case of manned systems, this integration has improved the ability of a single person to accomplish several tasks during the course of the mission. This concept, known as "integrated avionics" first started to appear in new aircraft such as the US Air Force F-22 and the commercial transport aircraft, Boeing 777.

As a result, and with an eye to reducing costs, the desire to extend the concepts of integrated avionics to vehicle management systems is totally subtle. This is borne out, moreover, by the great survivability, reliability and above all, availability of vehicle management systems. This result is to be found in the present report.

In so far as concerns the availability and reliability of the essential functions of vehicle management systems, as with flight control, extreme measures have been taken to guarantee restored operation in the event of failure. Nonetheless, these new technologies increase the cost of design, development and maintenance.

The new ideas summarised in this report also offer broader coverage of technical failures and lower design costs.

To date, all the problems have not yet been fully resolved. There is, however, an increased tendency to look for more safety and fault tolerance at lower cost. The recommendations given in Chapter 7 can only help to further reinforce NATO's aerospace supremacy.

Systèmes de gestion de vecteur intégré

(AGARD AR-343)

Synthèse

Les orientations majeures en matière de technologie, les objectifs dans les performances des systèmes d'armes et élaborer des systèmes aérospaciaux plus accessibles en les rendant moins onéreux, se révèlent des questions actuelles. Au sein de systèmes électroniques de tels concepts peuvent être simplifiés en tenant compte, plus particulièrement, des issues suivantes:

- La performance des systèmes d'armes;
- La technologie;
- La diminution des coûts.

Ce rapport s'intéresse essentiellement à une intégration des orientations définies ci-dessus: une intégration de la technologie afin de fournir des systèmes électroniques des haute performance à des prix accessibles. Beaucoup de questions trouvent une réponse cohérente et mesurée, dans le domaine de la conception et du développement. Le désir d'intégrer de façon fonctionnelle des sous-systèmes afin de réaliser de meilleures performances, a été amplement aidé par les découvertes en informatique. Pour les sytèmes pilotés, cette intégration a promu la capacité d'une personne à accomplir plusieurs tâches au sein d'une même mission. Ce concept, appelé les "avioniques intégrés", a commencé à apparaître dans les nouveaux aéronefs tels que l'avion de combat F-22 de l'Armée de l'air des Etats-Unis et le Boeing 777, avion de transport commercial.

Par conséquent, dans un souci d'économie, l'idée d'étendre les concepts d'avionique intégré aux systèmes de gestion des véhicules est tout à fait subtile. La preuve en est faite, entre autres, par la grande fiabilité, la confiance et par-dessus tout la disponibilité des systèmes de gestion des véhicules. Ce résultat est l'objet de ce compte rendu.

Pour ce qui concerne la disponibilité et la fiabilité des fonctions essentielles des systèmes de gestion des véhicules, comme le contrôle du vol, des mesures extrêmes pour garantir tout problème technique ont été prises. Néanmoins, ces nouvelles technologies accroissent le coût de conception, de développement et de maintenance.

Les nouvelles idées résumées dans ce compte rendu proposent, également, de fournir une couverture plus large des défaillances techniques et des moindres coûts de conception.

A ce jour, les problèmes ne sont pas totalement résolus. Les tendances à rechercher davantage de sécurité et de tolérances d'erreurs à moindre coûts s'accélèrent cependant. En procédant selon les recommandations mentionnées dans le chapitre 7, la suprématie aérospatiale de l'OTAN ne s'en trouvera que renforcée.

Contents

Page

Members of the Working Group

Co-Chairman:
Dr. Thomas B. Cunningham
Honeywell Technology Center
Minneapolis, Minnesota, United States

Co-Chairman:
Prof. John T. Shepherd
GEC-Marconi
Stanmmore, United Kingdom

Members

Dr. Chun Ho Lam
Principal Engineer
Allied Signal Aerospace Canada
255 Attwell Drive, Etobicoke
Ontario M9W 6L7, Canada

Mr. Hans-Jurgen Kaul
Head, Flight Guidance and Control
Deutsche Aerospace, Postfach 801160
D-81663 Munich 80
Germany

Mr. Hero Luero
Deutsche Aerospace
Postfach 801160
D-91633 Munich 80
Germany

Dr. Peter Lux
Dornier GmbH, EM3
Postfach 14 20
D-88039 Friedrichshafen 1
Germany

1st Lt. Fivos Hatzivasiliou
HAF Research & Technology Center
13, Solonos St
Kilkis 611 00, Greece

Mr. Massimo Avalle
Alenia Aeronautica
Corso Marche 41
10110 Torino, Italy

Ir Pieter Ph van den Broek
Department of Aerospace Engineering
Delft University of Technology
Kluyverweg 1, 2629 HS Delft
Netherlands

Maj. Ing. L.M. Alenquer
CLAFA/DE
Avenida da Forca Aerea
Alfragide, 2700 Amadora
Portugal

Engineer Antonio L. Alves-Vieira
Instituto Politecnico de Setubal
Escola Superior de Tecnologia
Rua do Vale de Chaves-Estefanilha
2910 Setubal, Portugal

Mr. Julian Simon Calero
Director, Div Propulsion y Energia
INTA, Carreta de Torrejon a Ajalvir
Km 4
28850 Torrejon de Ardoz
Madrid, Spain

Dr. Jose Maria Gargera Torres
Investigador, Division de Propulsion y Energia
INTA, Carretera de Torrejon a Ajalvir
Km4
28850 Torrejon de Ardoz
Madrid, Spain

Mr. Gordon Belcher
GEC Marconi Avionics Ltd
Airport Works, Rochester
Kent ME1 2XX
United Kingdom

Mr. Dwain A. Deets
Acting Chief, Research Engineering Division
NASA, Dryden Flight Research Center
PO Box 273, Edwards, CA 93523-0273
United States

Dr. Antony P. DeThomas
Wright Laboratory
Wright-Patterson AFB
5724 Pennywell Dr.
Dayton, OH 45424
United States

Mr. Donald E. Dewey
Associate Director of Technical Education
Boeing Commercial Airplane Co.
MS 13-16, Seattle, WA 98124
United States

Mr. Kevin R. Driscoll
Honeywell Technology Center
3660 Technology Drive
Minneapolis, MN 55418
United States

Dr. John Niemela
US Army CECOM
Chief-Command & Control System Div
Command, Control & Systems Integration Directorate
ATTN: AMSEL-RD-C2-ES
Fort Monmouth, NJ 07703-5000
United States

Dr. Edwin B. Stear
Corporate Vice President
The Boeing Company, PO Box 3707
Mail Stop 13-43
Seattle, WA 98124-2207
United States

Mr. Martin H. Stieglitz
Chief Engineer, RAH-66 Program
Boeing Defense & Space Group
PO Box 16858, MS P31-16
Philadelphia, PA 19142-0858
United States

Acknowledgments

The Integrated Vehicle Management Systems Working Group and this report are dedicated to the memory of its creator, the late Professor John T. Shepherd. The vision for such systems, which are currently being developed for numerous aerospace applications, was John's alone in 1988. This and numerous other contributions to guidance and control technology will last long after his departure in 1991.

The preparation of this report of Working Group 1 would not have been possible without the help of many individuals and institutions whose efforts are greatly appreciated. Leading experts in the fields of activities related to the deliberations of the Working Group contributed to the endeavor. Contributions consisted of presentations, discussion of ideas, and preparation of written materials.

Special thanks are given to the companies and institutions that supported the Working Group by providing support for their personnel to travel to a meeting and present applicable ideas and supporting information. These special presentations and discussions helped members of the Working Group better understand the state of the art, technology gaps, and future trends of the technology.

Appendix A of this report recognizes the contributions made by Working Group members and those made by invited guests.

List of Acronyms

AAP	Advanced Avionics Packaging
ABCCC	Airborne Battle Command and Control Center
ADIO	Analog and Discrete Input/Output
ADIRS	Air Data Inertial Reference System
ADIRU	Air Data and Inertial Reference Unit
ADS	Air Data System
AFDS	Autopilot Flight Director System
AIM	Actuator Interface Unit
AIMS	Airplane Information Management System
AIS	Airplane Information System
AMEL	Active-Matrix Electroluminescent
AMLCD	Active-Matrix Liquid Crystal Display
APS	Auxiliary Power System
APU	Auxiliary Power Unit
ASIC	Application-Specific Integrated Circuit
ATF	Advanced Tactical Fighter
AVIC	Air Vehicle Interface and Control System
AVMS	Advanced Vehicle Management System
BIT	Built-in Test
CATA	Control Allocation and Task Allocation
CBIT	Continuous Built-in Test
CCV	Control Configured Vehicle
CIP	Central Integrated Processors
CMAM	Custom Monolithic Analog Microcircuit
CMS	Cabin Management System
CNI	Communication, Navigation and Identification
COTS	Commercial off the Shelf
CPU	Central Processing Unit
DESC	Defense Electronics Supply Center
DDV	Direct Drive Valve
DFCS	Digital Flight Control System
DLA	Defense Logistic Agency
DMPG	DoD Microcircuit Planning Group
EAP	Experimental Aircraft Program
EC	Engine Controller
ECS	Environmental (or Engine) Control System
ECU	Environmental Control Unit
EFA	European Fighter Aircraft
EHV	Electrohydraulic Valve
ELMS	Electrical Load Management System
EMC	Electromagnetic Compatibility
EMD	Engineering and Manufacturing Development
EMI	Electromagnetic Interference
EMP	Electromagnetic Pulse
EOTADS	Electro-Optical Target Aquisition and Designation System
EUROCAE	European Organization for Civil Aviation Electronics
FAA	Federal Aviation Administration
FADEC	Full Authority Digital Engine Controller
FAST	Facility for Avionics System Test
FBW	Fly By Wire
FCS	Flight Control System
FDR	Flight Data Recorder
FH	Flight Hour
FLCS	Flight Control System
FMEA	Failure Mode Effects Analysis
FMS	Fuel (or Flight) Management System
FTIT	Fan Turbine Inlet Temperature

G&C	Guidance and Control
GLA	Gust Load Alleviation
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
GPWS	Ground Proximity Warning System
HIRF	High-Intensity Radio Frequencies
HLCS	High-Lift Control System
HOL	High-Order Language
HUD	Head-Up Display
HW	Hardware (or Hard Wired)
I/O	Input/Output
IAR	Integrated Avionics Rack
IBIT	Integrated Built-in Test
ICECS	Integrated Environmental Control System
ICLE	Integrated Control Law Evaluation
ICNIA	Integrated Communication Navigation Identification Avionics
IFIP	International Federation for Information Processing
IFFC	Integrated Flight Fire Control
IFPC	Integrated Flight Propulsion Control
IMS	Inertial Measuring System
IMU	Inertial Measurement Unit
INEWS	Integrated Electronic Warfare System
INS	Inertial Navigation System
IEC	International Electrotechnical Commission
IRS	Inertial Reference System
IRST	Infrared Search and Track
IVSC	Integrated Vehicle Subsystem Controller
IVMS	Integrated Vehicle Management System
JAA	Joint Airworthiness Authority
JIAWG	Joint Integrated Avionics Working Group
LCC	Life-Cycle Cost
LCD	Liquid Crystal Display
LRM	Line Replaceable Module
LRU	Line Replaceable Unit
LSI	Large-Scale Integration
LUV	Lethal Unmanned Vehicle
MC	Mission Computer
MDP	Mission Display Processor
MEP	Mission Equipment Package
MIPS	Millions of Instructions per Second
MOPS	Millions of Operations per Second
MPD	Multi-Purpose Display
MTBF	Mean Time Before Failure
MTBUR	Mean Time Between Unscheduled Repairs
MTTR	Mean Time To Repair
NOE	Nap-of-the-Earth
NRZ	Non Return to Zero
OBIGGS	On Board Inert Gas Generating System
PCS	Propulsion Control System
PFC	Primary Flight Computer
PFCS	Primary Flight Control System
PIBus	Peripheral Interconnect Bus
PICC	Processor Interface Controller and Communications
PIO	Pilot-Induced Oscillations
PS	Power Supply
PSA	Power Supply Assembly
PSC	Performance-Seeking Control
PWM	Pulse Width Modulation
RAM	Random-Access Memory
SAIRU	Standby Attitude Inertial Reference Unit
SEU	Single Event Upset
SMA	Stress Margin Approach

SMS	Stores Management System
SPS	Secondary Power System
SRU	Shop Recallable Unit
SSPC	Solid-State Power Controller
STOL	Short Takeoff and Landing
SuIT	Subsystem Integration Technologies
TOGW	Takoff Gross Weight
TRN	Terrain Referenced Navigation
TTL	Transistor-Transistor Logic
UMS	Utility Management System
UPS	Uninterruptible Power Supply
VFR	Visual Flight Rules
VLSI	Very Large Scale Integrated Circuit
VIU	Vehicle Interface Unit
VMP	Vehicle Management Processor
VMS	Vehicle Management System
VRU	Vehicle Reference Unit
V&V	Verification and Validation
WG	Working Group

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 INTRODUCTION AND BACKGROUND

Modern aerospace systems are required to perform greatly expanded mission functions with higher survivability rates at reduced costs. Such conflicting goals have stressed technology in important ways. While the explosion of digital and sensor technology has offered impressive performance enhancements for military weapons platforms, the costs of implementation and life-cycle ownership can be enormous if not carefully designed.

Fortunately, with the trends toward physical integration using modular avionics, we are starting to realize the reduced costs necessary. Unfortunately the benefits of integration have not been exploited by the vehicle management systems. The critical issue is to create these benefits yet guarantee the flight-critical and safety requirements of vehicle management systems.

This report deals with the design of the important features and functions of vehicle management systems necessary to achieve the benefits of integration.

1.2 VEHICLE MANAGEMENT SYSTEMS

For purposes of this examination, the functions of a given aerospace vehicle is broken into two distinct classes:

1. Payload functions,
2. Vehicle management functions.

Payload functions should be thought of as functions that relate directly to the mission of a given vehicle. For a fighter aircraft, these might be:

- Surveillance,
- Target detection and tracking,
- Identification—friend or foe (IFF),
- Fire control,
- Mission communications,
- Sensor control/sensor fusion.

Vehicle management on the other hand *enables* the vehicle to perform the mission function or supports the payload. The VMS is the collection of functions required for the vehicle to understand, plan, control, and monitor its operations. Once again, for a military fighter, these functions would include:

- Flight control (including FMC, GLA, ride quality, and others);
- Propulsion control;
- Flight-path commands;
- Multifunction and integrated navigation;
- Air data;
- Fuel systems;
- Electric utilities and power;

- Environmental control systems;
- Vehicle health monitoring.

Many of the VMS functions are essential to safe operations of the vehicle (i.e., includes all of the flight-critical and safety-related functions). This makes certain VMS functions subject to rigorous fault-tolerant and integrity design philosophies and subsequent critical safety of flight evaluations.

The nature of functions dealt with in this report, therefore, can be summarized in Figure 1-1.

It is not always clear which functions fall into the VMS versus the payload categories. For the fighter example, the following functions can be placed in either category:

- Integrated fire/flight control,
- SMS,
- Navigation,
- Mission planning.

Despite the controversy, one definition should be established. Any function that is flight or safety critical is part of the VMS. For instance, navigation can be included in payload definition or VMS; however, modern strapdown navigation systems typically share sensors with flight control functions. In this case, navigation would be a VMS function, as defined by Figure 1-1.

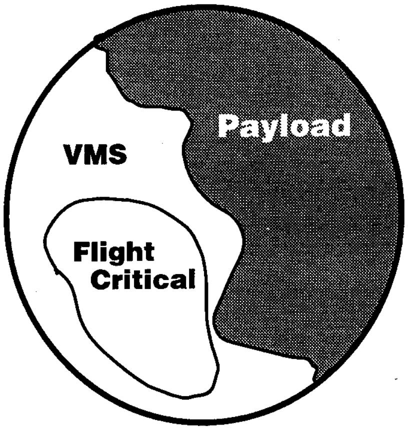


Figure 1.1. Vehicle Functions

These definitions apply to a number of vehicle types. Important examples are:

- Airplanes
  - Military fighters and transports
  - Commercial aircraft-transports, business, and commuters
- Rotorcraft
  - Spacecraft
  - Satellites and space stations
  - Launch vehicles
- Missiles
  - Strategic
  - Tactical-ground and air launched
- Other-Airship

VMS and payload functions for these vehicles are shown in more detail in Table 1-1. Although safety requirements for these vehicles vary considerably, the use of redundant systems and fault-tolerant design techniques is increasing in popularity for all these vehicle classes for economic reasons. The proper design of integrated systems is, therefore, of high interest for all of these vehicles. Notice that most, but not all, of the flight-critical functions are in the VMS category. In the future, all flight- and safety-critical functions should be contained under the VMS "fault-tolerant umbrella."

Table 1-1. Classification of Functions

Tactical Aircraft	VMS Functions	Payload Functions	Unassigned Functions
Safety/Flight Critical	Subsystem communications • Data buses • Processing network buses Utility systems • Fuel management and CG • Electrical power • Actuators • Landing gear • Hydraulic power Engine controls Flight control • Displays and controls • Sensors and compensation Life support/crew escape	Flight path management TF/TA sensors/control Stores arm/safe	
Mission Critical	Utility systems • Onboard health monitoring • Air data • Environmental control Navigation system • Inertial • TACAN • VOR/ILS/DME • LF and VHF direction finder	Mission processing system Subsystem communications • Data buses • Processing network buses Global memory Signal processors Graphic processors Communications Displays and controls Sensors EW Stores control Tactical situation manager	Flight data recorder

Missile	VMS Functions	Payload Functions	Unassigned Functions
Safety/Flight Critical	Propulsion control Guidance system • Processor • IMU • Attitude sensors Electric power control Flight control Thermal control	Command destract Safe/arm	
Mission Critical		Warhead control Seeker Fuse	

Table 1-1. Classification of Functions (continued)

Helicopter	VMS Functions	Payload Functions	Unassigned Functions
Safety/Flight Critical	Utility systems • Fuel/ECS • Electrical system • Hydraulic power • Landing gear Flight control • Displays and controls • Sensors and compensation • IMU		
Mission Critical	Processing • Sensors I/F • Comm/NAV Helmet integrated display/ sighting	Communications • Antennas • ICNIA • Audio • Comsec unit Controls displays • MFDs • Keyboard • Display generator • Payload EW	Crash recorder

Unmanned Autonomous Vehicle	VMS Functions	Payload Functions	Unassigned Functions
Safety/Flight Critical	Flight control • Processor system • Autonomous safe flight • IMU • Engine control • Actuators • Air data Electrical control Communications • Ground command RE link • Aircraft status to ground RF link Landing system • MLS • Landing gear		
Mission Critical	Onboard health monitoring Crash recorder IFF	Mission system	

Table 1-1. Classification of Functions (concluded)

Satellite	VMS Functions	Payload Functions	Unassigned Functions
Safety/Flight Critical	Electric power • Solar array • Batteries • Power control Thermal control Attitude control • Processor • IMU • Reaction control/CMGs • Star tracker/sun sensor		
Mission Critical		Payload sensor system	

In addition to vehicle types, the expansion of VMS functions in time has increased drastically. Table 1-2 shows this growth for helicopters developed at Boeing.

Table 1-2. Growth in Vehicle Management Systems at Boeing [1.1]

VMS Technology	Program				
	Tactical Aircraft Guidance System (TAGS)	Heavy Lift Helicopter (HLH)	Advanced Digital Optical Control System (ADOCS)	Osprey (V-22)	Comanche (RAH-66)
Explicit Model Following	x	x	x	x	x
Multi-mode Control Laws	x	x	x	x	x
Advanced Redundancy Mangm't	x	x	x	x	x
Sidestick Controller	x	x	x		x
Integrated FCS/Naviagtion			x	x	x
Integrated FCS/Engine				x	x
Integrated Diagnostics				x	x
Integrated Fire & Flight Control					x

1.3 BENEFITS OF INTEGRATION

Benefits of integration come in two ways: functional and physical. Functional integration most often benefits the operational performance of an aerospace application. Integrated fire and flight control is designed to improve the weapon delivery accuracy for a manned aircraft. Integrated flight and propulsion control is designed to improve the overall effectiveness of the vehicle/engine function. Functional integration is not the major objective of this report.

Physical integration has been demonstrated to provide effective delivery of aerospace functions while achieving benefits in lower development and recurring costs of avionics. Benefits often cited are:

- Lower design costs of a single (or few) common system(s),
- Lower recurring costs of extension and maintenance,
- Lower repair costs and higher sortie rate due to LRU replacement effectiveness,
- Lower spares costs,
- Lower purchase prices due to high volume and multiple vendors of common parts.

The magnitude of these benefits vary for different aerospace vehicles. The most pervasive example of physical integration is modular avionics.

1.4 MODULAR AVIONICS

One of the most exciting revolutions in aerospace vehicle electronics has been the implementation of digital systems with common electronic modules. The design philosophy is one of integrating functions in powerful computation modules of like designs. Figures 1-2 and 1-3 illustrate this trend. Figure 1-2 depicts the federated "black box" approach to avionics. Although some standards exist, the 1553B data bus, for example, the design of each box has been one of collaboration between the avionics (black box) supplier and the procuring agent for the particular function. For aircraft systems, this produced literally hundreds of unique design concepts with thousands of unique parts. The cost for such individuality is significant.

Integration, shown in Figure 1-3, involves significant standardization. The modules shown incorporate numerous functions such as display processing, central maintenance, and flight management. A number of programs have developed this technology, including PAVE PILLAR and PAVE PACE in the United States and the AAAP study in the UK. France and Germany have similar developments in this area. These systems benefit from impressive cost savings through the use of commonality of components and design. The benefits of are impressive.

For the F-22 these are:

- Fifty percent savings in maintenance cost;
- Increases avionics mean-time-between-failure (MTBF) rate by 400%.

These benefits are more recently being exploited by the commercial transport community. For the new Boeing 777 commercial transport, for example, projected benefits of integrated avionics are:

• Cost per function improvement:	70%
• Reliability improvement:	80%
• MTBUR/MTBF ratio goal:	90%
• Dispatch reliability target:	99%

Figure 1-3 also illustrates modular avionics and conventional "black box" avionics in a mixed architecture. This is typical of the current designs. The mission systems area are designed using avionics modules, whereas the VMS and some other functions are not integrated. Delineating the desire, design issues, and benefits of integrating *all* avionics functions, including the VMS, is the purpose of this report.

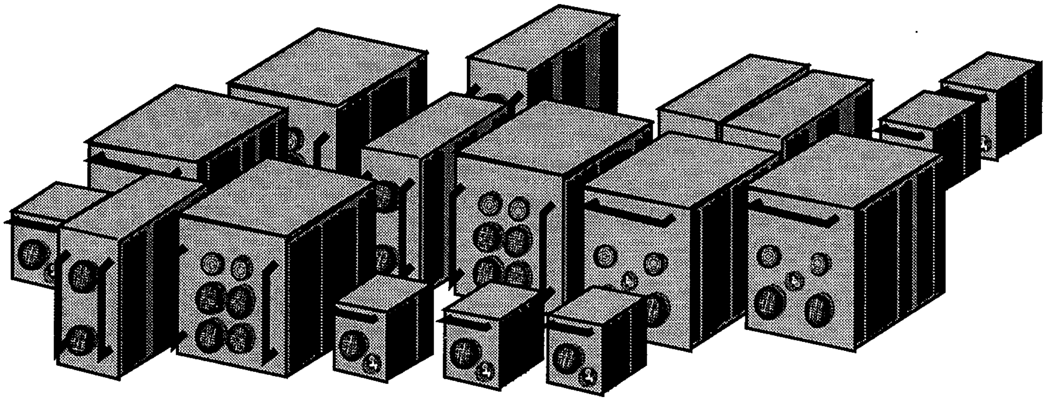


Figure 1-2. Conventional "Black Box" Avionics

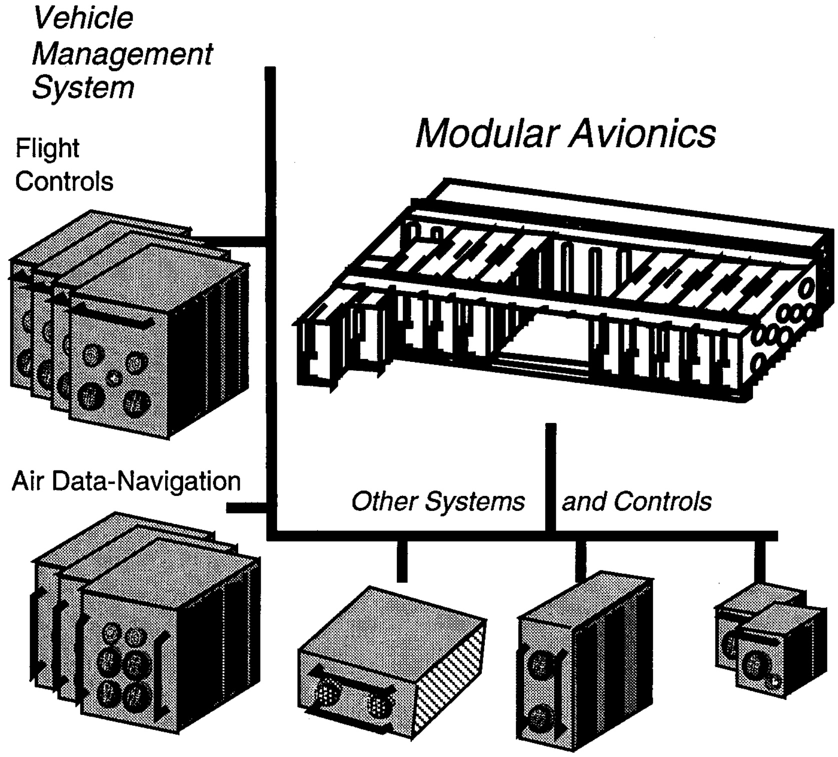


Figure 1-3. Mixed Integrated Avionics, VMS, and Other Subsystems

1.5 INTEGRATED SYSTEMS AND FAULT TOLERANCE

The flight control designer is keenly aware of a fundamental difference between flight control and other avionics functions; a much higher degree of fault tolerance and integrity. This has been achieved by a number of methods:

- Simple verifiable designs;
- "Brick walls" between FCS and other functions;
- Thorough verification, validation, and certification procedures.

Aerospace vehicles have benefited in the last decade from fly-by-wire design technology and digital systems used in flight control designs. These benefits have been achieved at the expense of simplicity of design and strained V&V and certification procedures. One area of major concern when using digital technology for flight-critical functions is the so-called "generic design error." The use of digital computers results in calculation paths that are difficult to guarantee under all circumstances.

In addition to concerns over unpredictable branching among various hardware components in memory and processor elements, the concern over software design errors is significant. These concerns have led to extreme approaches to software development:

- "N-version" developments consisting of totally independent channels of software starting with control requirements through systems V&V and certification;
- Formal methods to provide proof of fault tolerance for software.

These concepts are most critical in the development of Integrated VMS and are covered in detail in Chapters 4 and 6.

In general, the functional and physical characteristics of non-VMS and VMS systems are quite different. Table 1-3 describes this.

Table 1-3 introduces three critical issues related to integration and VMS:

- Functional sharing for VMS and non-VMS designs are different;
- Dissimilarity is used extensively in VMS designs;
- The sharing boundary between VMS and non-VMS systems is treated very carefully.

Table 1-3. Functional and Physical Commonality

Commonality Mode	**VMS**	**Sharing Boundary***	**Non-VMS**
Functional	Carefully done to insure integrity	------> <-/--/--	Unrestricted sensor sharing across functions
Physical	Current dissimilarity trend precludes like replication	------> <-/--/--	Multiple functions in each processor

* Data can flow freely from VMS functions to Non-VMS functions, but VMS functions must receive data very carefully

Ideally, we would like to create "seamless" sharing across the VMS/non-VMS boundary. Design methods must be developed to allow sharing. Even more controversial is the use of dissimilarity in VMS designs as a method of guaranteeing fault tolerance and integrity. Dissimilarity is the opposite of commonality and therefore creates a dilemma for integrated vehicle management systems (IVMS) based upon integrated modular avionics concepts. This dilemma is the key issue of this report and will be discussed thoroughly in Chapters 2 and 4.

1.6 INTEGRATED VEHICLE MANAGEMENT SYSTEMS

Integrated vehicle management systems can be thought of as the VMS functions implemented with modular avionics concepts. The critical issue for design of an IVMS is to exploit the benefits of modular avionics systems concepts while preserving the integrity required for flight-critical functions.

1.7 REPORT OUTLINE

This report will discuss and examine the key issues of integrated vehicle management systems. These issues are presented in the report as follows:

- Chapter 2 discusses integration concepts. Key features of physical versus functional integration, fault tolerance, and robust partitioning are described.
- Chapter 3 explains the benefits of integration for numerous examples.
- Architectures and implementation concepts are discussed in Chapter 4.
- Although no example of an integrated vehicle management system exists as yet, numerous examples of aerospace vehicles benefiting from current VMS design and integrated avionics are discussed in Chapter 5.
- As highlighted earlier, fundamental issues such as dissimilarity and software fault tolerance are barriers to integration for VMS systems. These issues and others require enabling technology advancements that are discussed in Chapter 6.
- Finally, recommendations for future developments are presented in Chapter 7.

1.8 REFERENCES

1.1 Landis, K. H., et al., Advanced flight control achievements at Boeing Helicopters, *Journal of Control*, Vol. 59, No. 1, January 1994.

CHAPTER 2

INTEGRATION CONCEPTS

2.1 INTRODUCTION

The necessity to balance the technology risk and the benefits related to the implementation of new functions naturally leads to the utilization of new technologies and design methods. The purpose is to amalgamate functions into hardware architectures and then systematically utilize integration in the development of vehicle systems.

Up to now, integration has been utilized in many non-flight-critical vehicle subsystems such as armament, navigation, and control and display. In flight-critical systems such as VMS and FCS, integration has not yet been fully exploited. The necessity to reduce weight and cost for all the vehicle systems leads to an introduction of new technologies enabling integration. In flight-critical systems, the introduction of these technologies must be balanced with the safety, redundancy, and survivability constraints.

The definition of IVMS supplied in Chapter 1 then leads to the need to clarify the relationship between the concept of integration and VMS.

Generally speaking, integration can be identified with amalgamation (or combination) of functions or subsystems that share a number of resources and supplying, globally, better performance than an equivalent nonintegrated system under some specific requirements and constraints such as: new function introduction, speed, weight, number of components, reliability, and maintainability. For example, the navigation and flight control subsystems can be integrated to provide an integrated autopilot function.

This report deals with system integration as opposed to device integration. Device integration is the process of increasing the functionality of individual electrical devices such as integrated circuits (ICs). While device integration is an enabling technology, it is not the focus of this report.

The two major types of integration are *physical* and*functional*. The first has as its aim the sharing of a single hardware device between some functions/subsystems, whereas the latter mainly uses communication cross coupling among subsystems with the aim of optimizing overall vehicle performance and reducing the pilot workload.

Integration types can be more clearly understood if described in a diagram such as Figure 2-1, where physical integration is shown as composed of three subcategories. Other ways to perform physical integration, although possible, have not been examined in this report. Chapters 2.2 and 2.3 will examine physical and functional integration.

Peculiarities and applicability of functional and physical integration, compared with a nonintegrated system, are shown in Figure 2-2. An example of physical integration is the common module avionics. A functional integration example would be integrated flight and fire control.

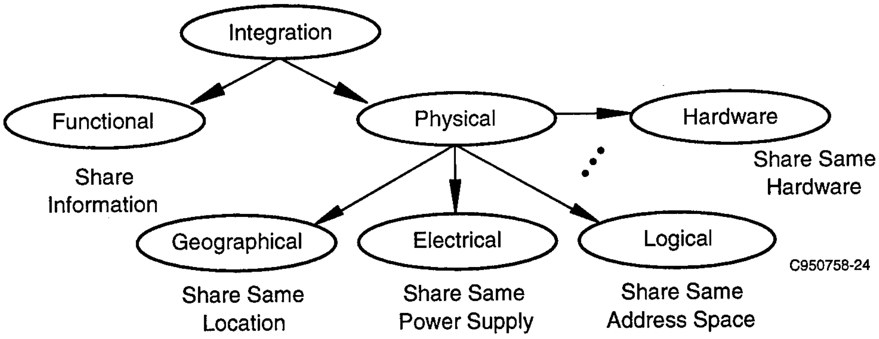


Figure 2-1. The Integration Tree

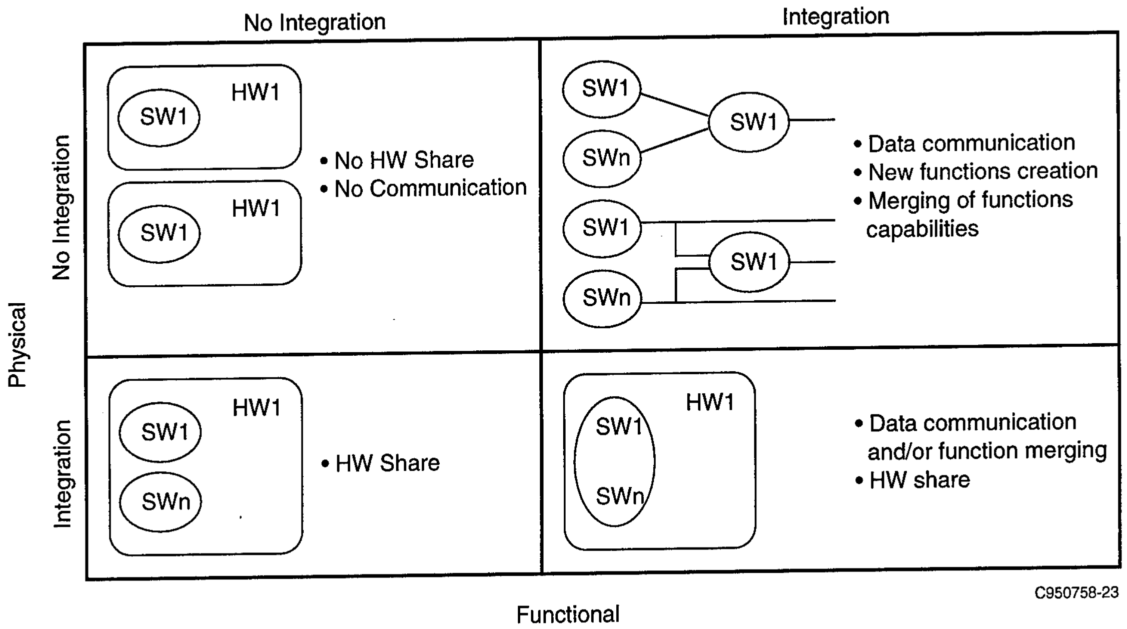


Figure 2-2. Functional and Physical Integration

Conceptually, the system development process usually starts with the definition and partitioning of the system features from a functional point of view. The next step is to choose an architecture able to contain such a system; then the functional architecture has to be mapped into the system architecture. As a consequence of such a methodology, it becomes evident that precise boundaries between physical and functional integration cannot always be easily defined and that gray areas usually exist. System designers have to take into account and balance common constraints and requirements.

The above items (the integration concept, methods to obtain integration, and implication for VMS) are discussed in the following subsections.

2.2 PHYSICAL INTEGRATION

The integration of a system from the physical point of view means to design a system as one "entity," based on the overall system consideration, with the objective of optimizing complete installation and reducing overall system costs.

Physical integration, that is:

- Sensor sharing (centralized air data system, dual use of IMU),
- Load sharing (processor share),
- Spare sharing (pooled spares).

provides means of increasing functionality within a reduced volume. Hardware, including sensors, required for the implementation of one subsystem is then utilized for the implementation of other functions and subsystems. In principle, the sensors necessary to meet the vehicle-related requirements are part of the VMS and those required to meet the mission-related requirements are contained in the avionics system. Both low-accuracy with high integrity and high-accuracy with low availability sensor data are needed in the VMS for guidance and control functions. An analysis of the functional requirements in the sensor area reveals a commonality of the VMS and other avionics systems for air data and inertial data.

A centralized air data system (ADS), for example, is part of the VMS due to the high integrity requirements imposed on that subsystem. The objective of the centralized ADS is to provide digital air data information to all vehicle and avionics systems requiring them and via the various display suites to the pilot. The ADS accuracy requirements are mainly driven by the avionics system (e.g. , controls and displays).

Development programs also verified that a multifunction inertial system can reliably serve as a single source of inertial data for the entire vehicle. However, a centralized inertial measurement system (IMS) is not always feasible. Structural effects can require measuring inertial data for avionics and VMS at different locations in the vehicle. In this case, a central IMS might not be weight and cost optimal. Therefore, a high-accuracy/low-availability system being part of the avionics and a medium/high-integrity system being part of the VMS could be chosen. The latter, however, should be accurate enough to provide the backup navigation function. Through physical integration, the VMS inertial measurement system can form part of the vehicle inertial measurement system.

Full physical integration requires that the VMS and its subsystems not be designed in isolation from the avionics system. Thus, both systems have to follow a common design method in which the functional and performance requirements are conducted in a phased manner following a similar design path. Avionic design requirements should provide insights into the appropriate levels of the IVMS design and vice versa. Sharing resources in a system implies at least a coordination between the users of such resources. Adequate documentation at the system level is required to define the subsystem-to-subsystem interaction and provide a basis for system level testing. Suitable traceability methods should provide the designer and the consumer of a service the means of verifying that the requirements placed on a system have been met. The impact of low-level design changes on the overall system design must be clearly visible.

2.2.1 Geographical Integration

Performing integration from the geographic point of view means, for example, putting two (or more) processors into a single box as shown in Figure 2-3B. In this case, the hardware reduction is at a very low level. It is limited to the reduction of the number of boxes with respect to Figure 2-3A, where no integration is performed.

2.2.2 Electrical Integration

Performing integration from the electrical point of view is a possible subsequent step with respect to geographical integration. In this case, the number of power supplies is reduced to one and the hardware reduction becomes more evident, as shown in Figure 2-3C.

2.2.3 Logical Integration

Integration from the logical point of view is mainly related to the computer science level. It can influence system integration (physical or functional), making available more powerful and efficient computers.

Logical integration shares a unified address space. This concept is sometimes called "tightly coupled multiprocessor processing." Taking into account such an introduction, it could be stated that the logical integration concept will mainly involve computer elements such as the following:

- Bus,
- CPU time,
- Memory and I/O addresses.

Physical integration is directly influenced by logical integration because it is strictly related to the computer's power and efficiency. Utilization of more powerful and efficient computers leads to the possibility of running more software in less time (more MIPS). Then more functions can be integrated into the same hardware, enabling physical integration.

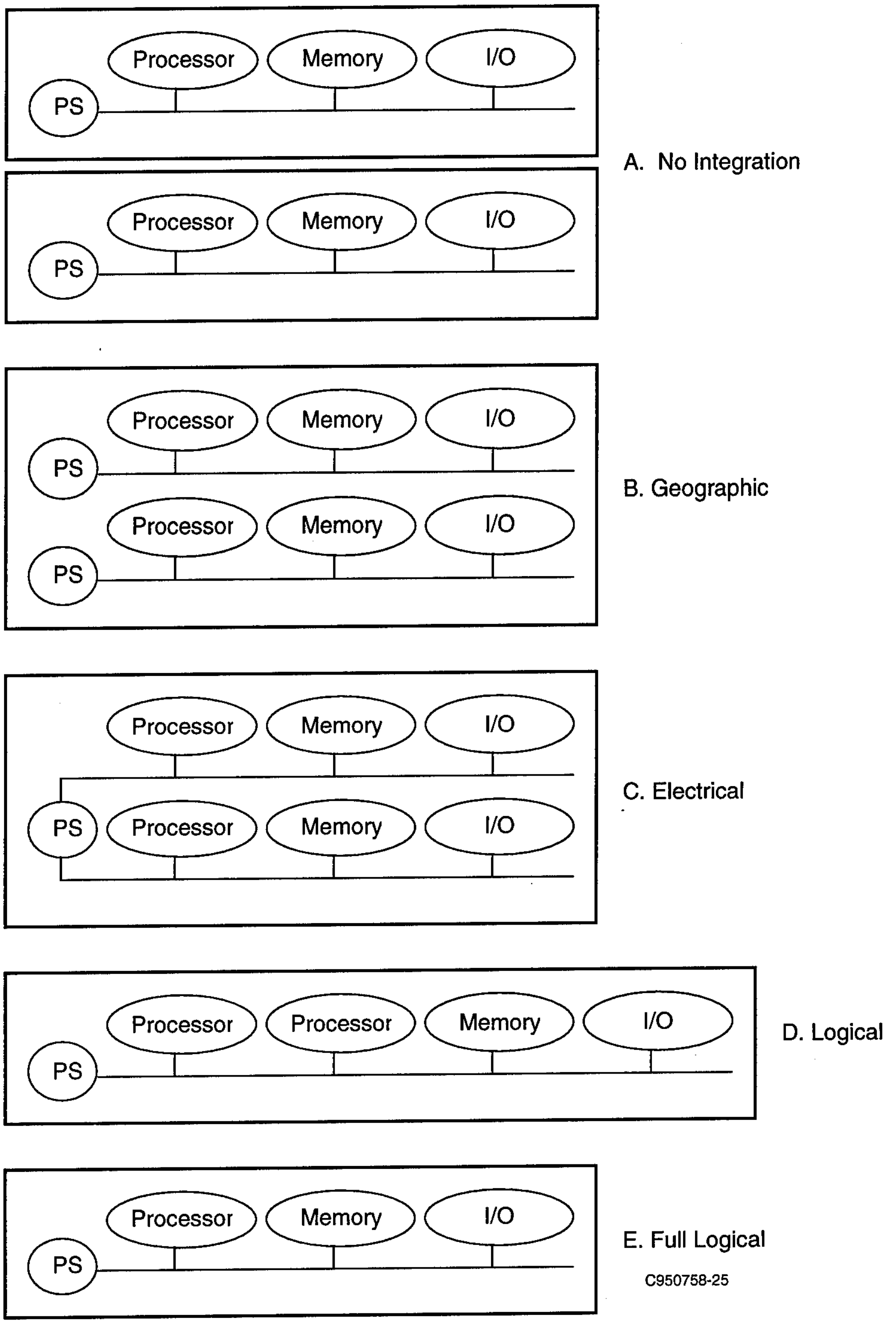


Figure 2-3. Principles of Physical Integration

Functional integration is influenced by logical integration as more efficiency in bus and memory utilization leads to better interfunction data exchange.

Figure 2-3D shows the hardware reduction achievable by integrating memories (shared memories) and I/O (shared I/O). Note that in this case, arbiter functions should be added to regulate the processors' access.

Figure 2-3E shows a more integrated system where the processors are integrated into a more powerful one.

2.3 FUNCTIONAL INTEGRATION

System integration from the functional point of view means to interconnect, via information or control signal exchange, two or more subsystems that are mutually compatible and coordinate their functions in a global sense to optimize overall vehicle system performance and to achieve a pilot workload reduction. The design for functional integration treats the entire vehicle as one dynamic system, measured by a variety of sensors and interfaced to the pilot.

Functional integration can further enhance performance and operational capabilities of a vehicle by using the large degree of dynamic cross-coupling among the subsystems. An effective integration can optimize the favorable interactions to enhance vehicle maneuverability, precise flight path control, and fault-tolerant system design.

Historically, vehicle design has been based on the philosophy that most (sub)systems (i.e., flight control and propulsion control) can be designed independently and that the pilot will serve as an integration medium and will effectively integrate the (sub)systems by its control inputs. With the increasing complexity and sophistication of new vehicles modes and control functions, coupled with reduced reaction times as vehicle speed and maneuver rates increase, the pilot workload can rapidly increase to unacceptable levels.

Many examples of vehicle systems already exist with varying degrees of functional integration. Control Configured Vehicles (CCV) have been developed by integrating the airframe with the stability augmentation system, and the performance and safety of the vehicle are dependent on an electronic fly-by-wire (FBW) system. Other typical examples of functional integration for high-integrity systems are automatic landing and terrain following. Several past and ongoing research efforts have addressed the issue of flight/propulsion integration. Advanced features such as four-dimensional navigation, optimum trajectories, and active control of engine surge margin in response to vehicle maneuver requirements can be realized through integration. An impressive sample is the STOL Maneuvering Technology Demonstrator [2.6]. Figure 2-4 illustrates this integration.

In particular, navigation can achieve very effective enhancements from functional integration, because of the highly specialized sensors and processors required to implement the basic navigation elements like INS and GPS. The diverse sensor set can, in turn, limit effective physical integration.

In Figure 2-5, the functional scheme of a navigation system enabling new features is shown. The system is based on four distinct and physically separated devices: a radar altimeter, an INS, a GPS, and a TRN [2]. It is known that the INS can ensure good short-term precision, but it becomes inaccurate in the long term due to gyro drifts. On the other hand, GPS has very high long-term accuracy but has poor short-term accuracy due to possible satellite obscuration. A first functional integration can be obtained by integrating INS and GPS using a Kalman filter to coordinate the long-term precision of GPS with the short-term precision of INS. In this way, a good knowledge of aircraft position can be maintained during the mission. The subsequent step is integration of the radar altimeter and TRN with INS/GPS, obtaining a more integrated and safe system that is able to follow a predefined flight path, integrating the knowledge of accurate aircraft position with the knowledge of the ground profile overflown by the aircraft. Such a functionally integrated system can then augment system performance with respect to a nonintegrated navigation system and reduce the pilot workload, because fixes can be automatically (and more frequently) performed by the TRN. Additional (functional) integration involving the air data system and FCS is necessary for development of such a system, as the development of flight director and autopilot functions supplying the pilot with proper indications about the flight conduction are mandatory to utilize such an integrated navigation system effectively.

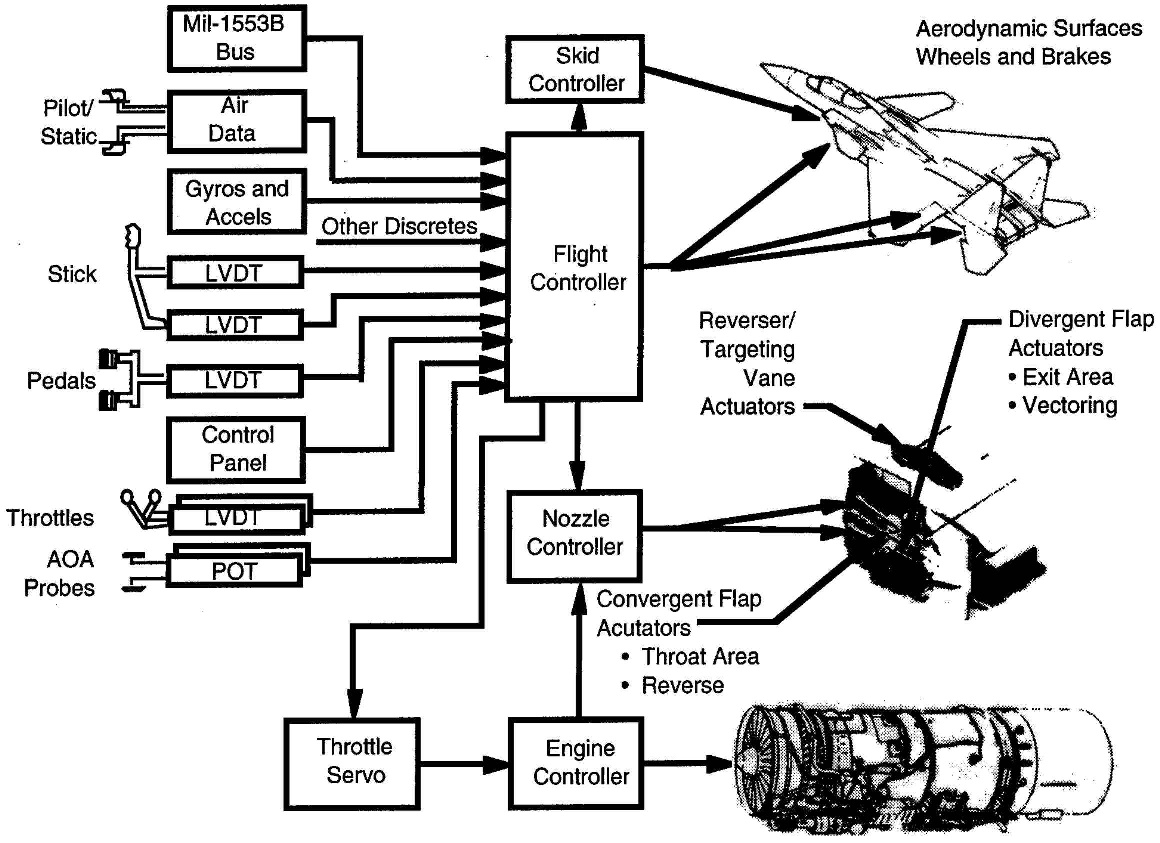


Figure 2-4. Integrated Flight/Propulsion Control [2.6]

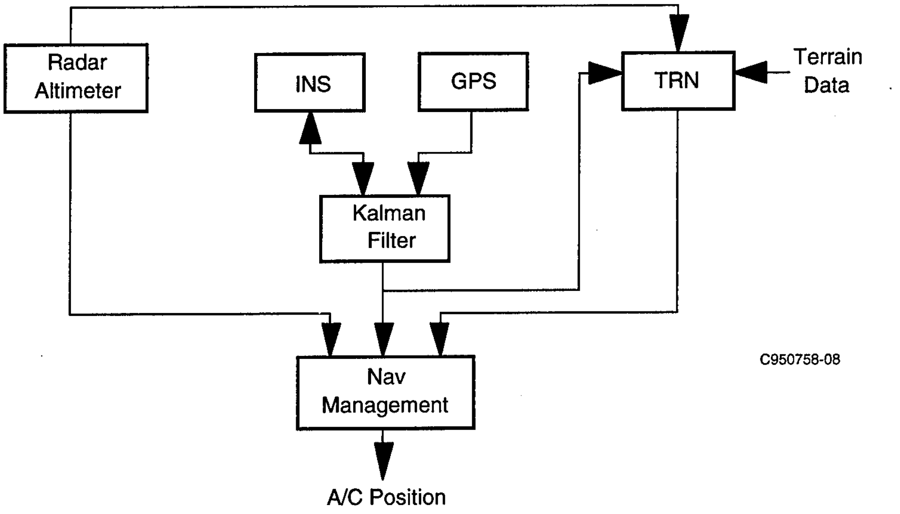


Figure 2-5. Integrated Navigation System

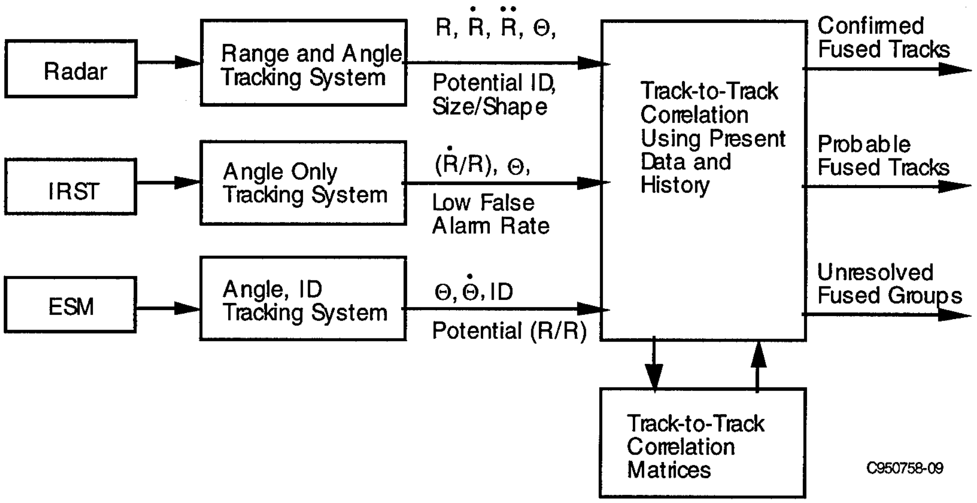

Radar
Range and Angle Tracking System
$R, \dot{R}, \ddot{R}, \Theta,$
Potential ID, Size/Shape
IRST
Angle Only Tracking System
$(\dot{R}/R), \Theta,$
Low False Alarm Rate
ESM
Angle, ID Tracking System
$\Theta, \dot{\Theta}$, ID
Potential (R/R)
Track-to-Track Correlation Using Present Data and History
Confirmed Fused Tracks
Probable Fused Tracks
Unresolved Fused Groups
Track-to-Track Correlation Matrices
C950758-09

Figure 2-6. Sensor Level Tracking

The above example shows that the introduction of new system features utilizing functional integration can sometimes lead to the necessity to implement other features necessary to utilize them. This leads to a growth in the system level of integration and system complexity. The impact on the system design could be very high, especially for the bus communication rate and computers computational load. Costs and benefits then have to be well balanced with the system requirements.

As the degree of functional integration between the various vehicle subsystems increases and consequently the pilot workload reduction becomes feasible, the role of the pilot should change. The pilot is becoming more of a flight manager, and his/her information needs are changing. The man/machine interface consideration should therefore be an integral part of the vehicle control system design.

2.4 DIFFERENCES BETWEEN FUNCTIONAL AND PHYSICAL INTEGRATION

At this point in the discussion, it should be realized that functional and physical integration are strongly related and system design utilizes features and benefits of both (see Chapter 3). On the other hand, differences between physical and functional integration are also evident and different problem approach philosophies can provide advantages of one method with respect to the other, leading to different system design solutions.

To clarify such a concept, again taking into account Figure 2-2, two examples of sensor fusion systems where integration is performed in different ways are now discussed. There are two approaches used by the sensor fusion community. These approaches are: sensor-level tracking, Figure 2-6, and central-level tracking,

Figure 2-7 [2.1]. The sensor-level tracking approach uses a function to fuse the single tracker's outputs into a common output. In this case, sensor fusion is obtained via functional integration, in the sense of Figure 2-2, as independent outputs from physically separated functions (the trackers) are used to obtain a new function, augmenting overall system performance and reducing pilot workload.

The central-level tracking approach performs sensor fusion following an approach completely different from the sensor-level tracking approach, as it does not integrate signals or information but integrates the tracking systems of the single sensors into a unique tracker directly performing sensor fusion. In this case, the physical and functional integration concepts are used together to build a sensor fusion system. It should be noted that the trackers of the single sensors are not simply added to form a unique tracker, but a single and completely new tracking algorithm, taking into account the characteristics of all the involved sensors, is used.

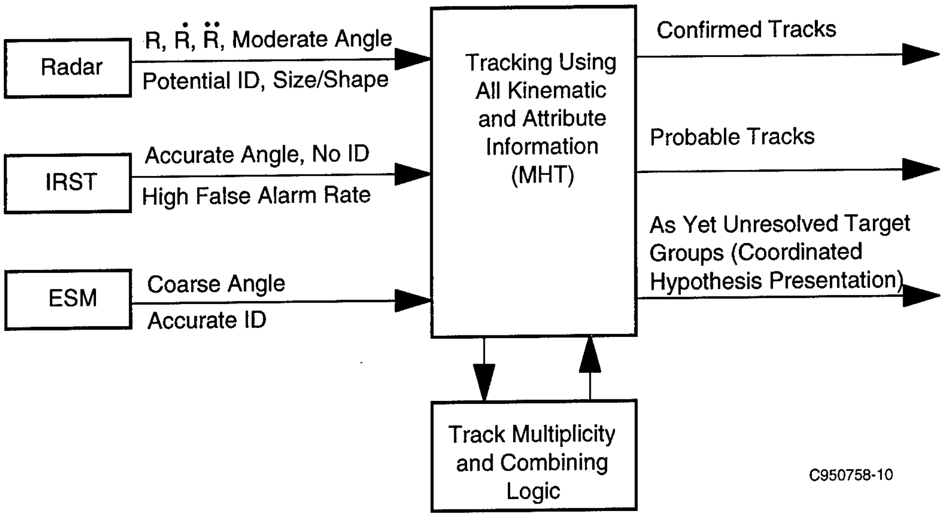

Radar
R, $\dot{R}$, $\ddot{R}$, Moderate Angle
Potential ID, Size/Shape
IRST
Accurate Angle, No ID
High False Alarm Rate
ESM
Coarse Angle
Accurate ID
Tracking Using All Kinematic and Attribute Information (MHT)
Confirmed Tracks
Probable Tracks
As Yet Unresolved Target Groups (Coordinated Hypothesis Presentation)
Track Multiplicity and Combining Logic
C950758-10

Figure 2-7. Central Level Tracking

Examples of pure physical integration applied to sensor fusion do not exist in the literature; therefore, this case is not discussed here as it would be purely academic.

In analyzing Figures 2-6 and 2-7, it is evident that the two systems are very different and their outputs have different sensor fusion characteristics. The sensor-level approach is generally more simple to implement, requires a medium to high computational load for the computers, and is more safe, as the loss of the sensor fusion computer just implies the loss of the sensor fusion function and not the loss of the target tracks. On the other hand, the precision of the fused tracks will simply be the best of the received sensors tracks.

The central-level approach can guarantee better precision and a reduced misassociation probability. However, a large amount of data must be transferred from sensors to the central tracer, as periodic track transmission requires less data transfer with respect to the sensors' observation transmissions (false alarms are still present). In addition to the above considerations, the system is less safe, as the loss of the tracking computer implies the loss of the system tracking capability.

Main differences between functional and physical integration can be extracted from the above discussion. It is evident that physical integration essentially produces a reduction in the hardware necessary to build the system (cost reduction) while functional integration usually requires new hardware to perform the integration of the outputs from systems. The hardware reduction obtained with physical integration has to be balanced with the spares necessary to meet the safety and integrity requirements. Likewise, functional integration automatically introduces some intrinsic redundancies in the system, introducing a cost growth, but can provide graceful degradation capabilities.

In conclusion, functional and physical integration should be balanced during system design on the basis of performance, cost, safety, and integrity requirements.

2.5 INTEGRATION OF MANAGEMENT SYSTEMS

Management systems such as VMS could become integrated systems (IVMSs), exploiting the features of both physical and functional integration described in the preceding subsections. Trends in the development of modern systems are toward the implementation of several control systems on vehicles. Then the integration concepts can be extended to the entire vehicle, at least by functional integration. The integration of management systems enables the reduction of pilot workload and can exploit the benefits of better coordination between the vehicle systems and of resource optimization.

The above considerations can open the way to a fully integrated vehicle in which interactions between high-level and philosophically different systems have to be carefully analyzed by the designer and the related costs/benefits have to be carefully weighted on the basis of vehicle operational requirements.

2.5.1 Automation in Management Systems

When vehicle systems are developed independently, the task of collecting information and taking actions is entirely the pilot's responsibility. If integration starts to involve management systems, the pilot will no longer be alone in the overall vehicle visibility, as management systems, via their integration, would have access to complete vehicle status. Functions could be checked easily to monitor variations.

The next step would be the introduction of automation in the management system. This would give it the capability of making autonomous decisions based on measured parameter status, thus relieving the pilot of some repetitive actions.

Specific actions or controls that could be transferred from the pilot to an automatic management system depend on the vehicle mission and on the level of integration between the management systems.

2.5.2 Management Hierarchy

Consider as an example the two main management systems present on a vehicle: mission and vehicle management. These systems are not totally independent but need at least some intervention from the pilot to perform their tasks.

Transferring some tasks from the pilot to the management systems increases the necessity to create an integrated automatic control to coordinate their actions. The idea of superimposing a controller on such management systems tends toward the Pilot Associate concept, which is not covered in this report.

A large central controller can be avoided by introducing hierarchy levels between the management systems based on their relative importance. The higher-level management systems are used to control the lower level systems. In this case, the flight manager controls the vehicle manager and the mission manager controls the flight manager. Such an idea simplifies the problem, reducing the impact on the system.

2.6 INTEGRATION CONCERNS

When performing system integration, there are performance enhancements and possible cost reductions. Both of these factors are influenced by technical and design choices related to different system developments and having different performance characteristics. As the creation of new integrated functions expands, the growth in design complexity has to be taken into consideration.

Design approaches and methodologies should then be developed to accommodate such new features and functionality requirements. The following subsections will analyze such new design features and functionalities.

2.6.1 Functional Partitioning

Once the system functional design has been performed, the next step is the embedding of functions into the hardware architecture. Particular care must be taken to perform this embedding, as it can directly influence system performance and, consequently, system requirements. Such an influence of hardware architecture (physical integration) on functional performance, and consequently on functional integration, again demonstrates the close relation of the two integration methodologies. As a simple example, clearly data exchange among functions can be performed more efficiently if functions are embedded into the same processor instead of into two separate processors connected via a data bus; however, the V&V activity becomes more challenging.

The following subsection will discuss a functional integration methodology that enables functional partitioning.

2.6.2 Grouping Functions

As described in the preceding subsection, correct functional partitioning into the hardware architecture must be performed during the system design process. The functional partitioning should enable functional integration, taking into account its main problems. Such a synergy between functional and physical integration can simplify overall system integration and consequently improve its benefits.

The main concept related to such a design methodology is the creation of functional areas (sometimes called "metafunctions") where functions assigned to common tasks are grouped. This function affinity can lead to an optimization of bus communications and consequently to the reduction of the load and of the data latency on buses.

Integration among functional areas, where usually bus latency should be minimized, can be achieved via high-speed data bus enabling fast communications.

In a system organized in a number of interconnected functional areas, there are two different levels of integration. They are *inside functional areas,* where the level of integration will be very high due to the affinity of the integrated functions, and *between functional areas,* where the level of integration will be medium to low, depending on their common tasks.

The assignment to group functions for integration purposes, on the basis of their affinity, also enables separation among safety-critical and non-safety-critical functions.

2.6.3 Reconfiguration

In integrated systems, the introduction of new functionalities may be feasible and cost-effective. In particular, reconfiguration is enabled by functional and physical integration of system functionalities and is currently applied to mission and sensors systems.

The purpose of reconfiguration is to reassign functions inside the physical architecture to optimize the computational power or other hardware utilization during the various mission phases or to guarantee the availability of essential functions after hardware failures.

Reconfiguration directly impacts the integration between functions: after a reconfiguration, new functions must guarantee integration with those areas of the system not involved in the reconfiguration, possibly without lack of synergism.

The performance desired after reconfiguration will directly influence the level of integration (from both the functional and physical points of view). After a failure, for example, an integrated system can try to maintain its performance by operating with a new configuration. If no spare resources are available to operate such a configuration, the system has to decide if some functions can be eliminated from the system or if they must continue to run with reduced performance.

The first case will cause a reduction of the integration level as the missed function will no longer contribute to overall system performance. In the second case, the level of integration can be kept with all the functions active, but the overall system will have reduced performance.

The reconfiguration concept, is not directly applicable to IVMS due to the time required to reload and restart the software routines into new processors; typical IVMS constraints concerning reconfiguration are discussed in Subsection 2.9.2.

2.7 FAILURE TOLERANCE, INTEGRITY, AND CRITICALITY

2.7.1 Shared Redundancy

The concept of failure tolerance should be modified as the integration level increases.

The intrinsic capability of an integrated system to optimize the number of resources, discussed in the previous subsections, could be useful when dealing with the system failure tolerance and system integrity constraints.

In a nonintegrated system, or in a system with a low level of integration, fault tolerance is usually implemented by duplicating some critical resources. In an integrated system, in some cases, duplications can be eliminated or reduced due to the availability of similar resources shared within the system that allows maintenance of system performance. In addition to hardware duplication, reconfiguration is also useful to optimize the performances after a failure recovery.

Physical integration can contribute to hardware spare reduction, as the total amount of hardware is reduced with respect to a nonintegrated system.

It is possible to again use the sensor fusion example of Figure 2-7 to explain such a concept. If target tracking has to be guaranteed for every sensor, even after a hardware failure, for the case of a nonintegrated system (i.e., in a system without sensor fusion), it is necessary to duplicate all the trackers to meet such a requirement. However, in an integrated system (i.e., a system with sensor fusion), only one processor has to be duplicated.

2.7.2 Criticality Level

When integrating and controlling management systems, the problem of multiple criticality levels arises.

Usually the flight management and part of the vehicle management systems are flight-critical, whereas the mission management system is mission critical. However, if the mission manager controls the flight management system, as in Subsection 2.6.2, it has to supply flight-critical data to the flight manager. The part of the mission management system generating such data should also be flight-critical unless the flight manager can detect erroneous data and take corrective action.

The mixing of critical and noncritical functions in a single processor, regardless of interaction, has not heretofore been done. New techniques based on robust partitioning have been developed to allow careful mixing of critical and noncritical functions [2.6].

2.7.3 Integrity

The flight-critical nature of the majority of vehicle functions dictates that the IVMS architecture meet unique design challenges. The computing system configuration must take into account the flight safety requirements of the entire vehicle, including sensors and actuators. The IVMS requirements will therefore be much more stringent and more constraining than those for non-VMS avionics functions.

The current design objective is that those aircraft losses due to hazardous technical failures in the IVMS alone shall not exceed 10^{-7}/FH for military aircraft and 10^{-9}/FH in case of commercial aircraft. This requirement is coordinated with the overall aircraft loss rate and coupled with the contributions of the other vital A/C systems such as engines, hydraulic supplies, etc. It seems valid to assume that this requirement will not increase for future vehicle developments.

The IVMS shall be based on the fault containment principle. It has to possess redundancy management techniques capable of providing optimum failure survivability via detection and isolation of failed components and reconfiguration of the remaining healthy components to provide the maximum level of vehicle safety and the highest probability of mission completion. Redundancy management strategies are presently almost exclusively directed at protection against random hardware faults, and fault avoidance techniques have been the main method of achieving high-integrity software.

Safe, dependable failure detection and identification schemes are necessary for detecting and isolating failures in IVMS sensors, data links, actuation systems, electrical power supplies, processors, data networks, and hydraulic power supplies. It must be accomplished in such a manner that it does not degrade the capability to detect and isolate subsequent failures. If a failure has been detected, reconfiguration to the next lower redundancy level of the affected function is performed and transients due to reconfiguration are acceptable. System degradation due to nuisance false alarms must be considered when system performance is being evaluated.

Various surveys of fault-tolerant computing introduce many of the concepts and definitions relevant to digital systems, but they need to be interpreted in the light of the IVMS application. Moreover, the surveys cited above do not all agree on the definitions. A consistent set of fault tolerance terminology should be adopted within the industry, as currently different definitions are used by the customer, airframe manufacturer, and equipment supplier (Section 2.10).

Several quadruplex and triplex redundant, fly-by-wire flight control system configurations have been designed, validated, and flight demonstrated during the last decade. The systems, which have an extensive record of performance to support critical applications are either minor-frame synchronized or asynchronous architectures. They are based on static redundancy, that is, comparison monitoring (cross-lane monitoring) as the principal method of failure detection and isolation. Self-tests (in-lane monitoring) are not the primary means of defense but are used to enhance the failure detection coverage in areas where defects might otherwise remain dormant in flight and to enhance the availability of secondary facilities (sensor or actuators) so as to improve overall system reliability.

Analytic redundancy can be used for either in-lane monitoring or to generate additional inputs to cross-lane monitors. Analytic redundancy is the use of relationships among dissimilar sensed phenomena to achieve fault detection and isolation. The use of analytically generated signals has been heavily researched in recent years and is aimed at reducing the cost of redundant hardware and improving vehicle survivability by allowing more dispersion of components. Analytical redundancy will only be accepted if the analytical model can be determined with sufficient accuracy (e.g., adequate knowledge of the sensor characteristics during a failure is available and correctly modeled).

The research community has developed several approaches to the implementation of software fault tolerance. The proposals that have received the most attention are N-version programming and recovery blocks. These methods, discussed in Chapter 4, Subsection 4.8, still face several practical difficulties in their implementation.

Hardware diversity ("dissimilar processors") and/or analog/digital backup system(s) with at least level 3 handling characteristics as a concept to circumvent the existence of design (generic) faults/errors places an additional burden on the development process. The advantages are that it diminishes the concerns that residual, undetected design errors could have catastrophic consequences. However, the gains cannot be quantified, as it is impossible to predict the probability of occurrence of generic errors. This approach can create more difficulties than it removes and can also lead to a relaxation of design discipline. The inclusion of a backup system is often based on emotional feelings and/or because the purchaser does not believe that the integrity of the software can be adequately demonstrated. Mechanical backup control systems might not be feasible in the case of a relaxed static stability aircraft. Although many have a phobia about generic software faults, software is no more prone to generic faults than a similarly complex system implemented as digital hardware, analog hardware, hydraulics, mechanics, etc.

2.8 LEVEL OF INTEGRATION

The number of functional or physical elements involved in integration can vary depending on both the specific application and design choices. They, in turn, can limit integration at part of a subsystem, to an entire subsystem, or to extend it to the overall system.

Figure 2-8A shows a traditional configuration of signal and data processing in which each sensor has its own signal processor and its own data processor fitted with specific software. In a system such as this, integration is at a very low level.

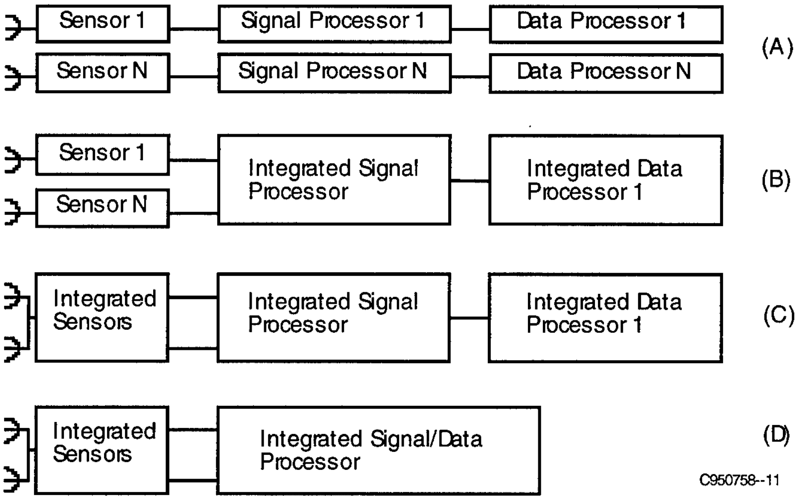


Figure 2-8. Levels of Integration

Figure 2-8B shows a system configuration in which integration involves signal processors and data processors, rising up the level of integration for the overall system.

Figure 2-8C shows a system in which integration also includes the sensors.

Figure 2-8D shows a system in which integration is at the top level and involves both the signal and data processors.

The above examples demonstrate the main difference between existing mature technology (Figure 2-8A) and new trends in system development (such as Pave Pillar) (Figure 2-8B).

It should be noted that available technology does not yet support the development of systems such as those shown in Figures 2-8C and 2-8D, but they are useful to understand the integration trends.

The utilization of mature technology to develop integrated systems can strongly influence the level of integration because classic boundaries between subsystems can limit integration possibilities. Thus, the level of integration achievable in a system strongly depends on the adopted technology and on the number of traditional subsystems being considered.

A high level of integration can also produce benefits by reducing the number of control resources that remain idle for long time periods (for example, undercarriage control).

Increasing the level of integration also implies the adoption of general-purpose processors able to serve different kinds of subsystems and functions.

In general, the higher the integration level for a system, the better will be the resource optimization. On the other hand, as the integration level for a system increases, the system complexity will also increase. Thus, integration should not be intended as a method of simplifying systems but as a way of optimizing complex systems. The integration level should be strictly related to the optimization concept: any system should have an optimization limit, and such a limit will influence the optimum integration level for that system.

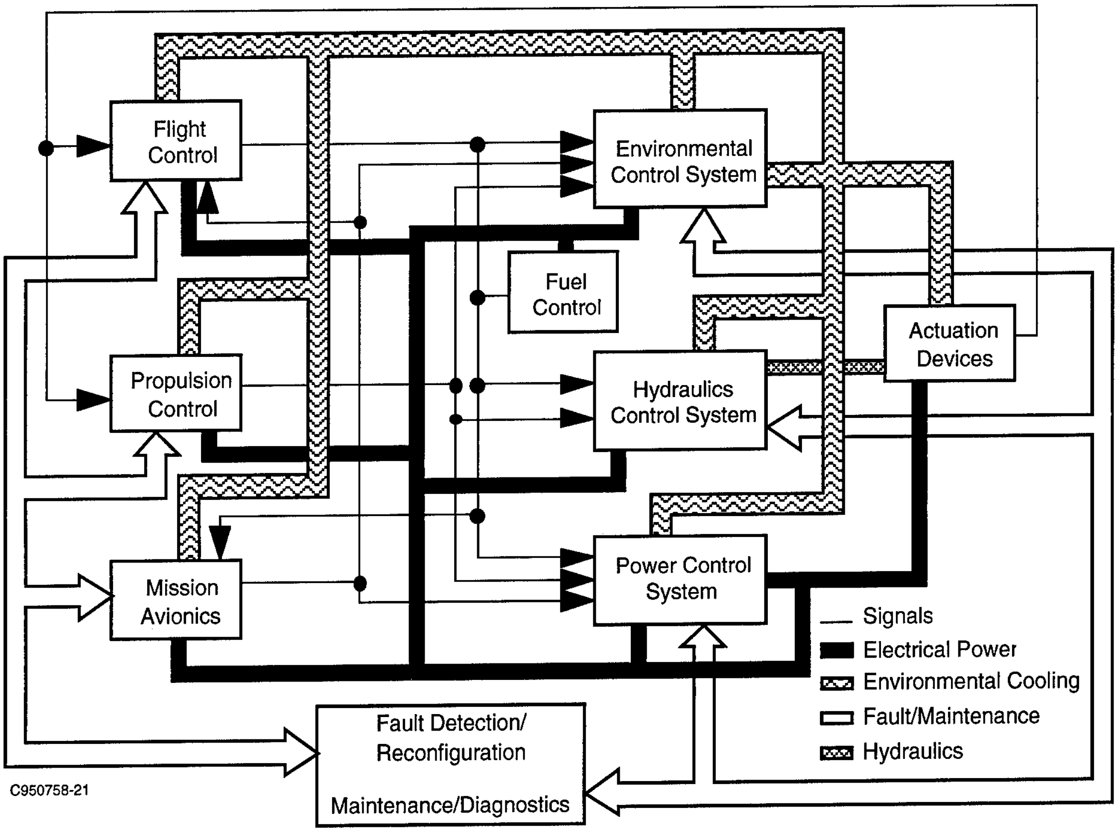


Figure 2-9. IVMS Functional Diagram

2.9 SPECIFICATION FOR IVMS

The specification of a VMS is a technology integration task. State-of-the-art assessments and trends in the underlying computing, sensing, and actuation areas must be performed to select from a number of design alternatives. The selection process of competing architectures is often the result of difficult compromises among numerous and sometimes conflicting requirements such as weight, volume, power requirements, survivability, maintainability, life-cycle cost, and others. This configuration design process is not currently supported by a technology capable of quantifying the relative merits of competing configurations.

The ultimate goal of the IVMS is to provide a system that leverages commonality and shared resources among the electronics associated with the control, monitoring, reconfiguration, and diagnostic processing of all vehicle-related functions. Maximization of common line-replaceable modules (LRMs) within and across systems will have a favorable impact on acquisitions, logistics, and life-cycle costs. Additional objectives are to enhance the overall performance or capability of the vehicle by means of integrated functions.

The IVMS will generally be composed of the following vehicle flight systems:

- Flight controls,
- Propulsion controls,
- Inertial and air data sensors,
- Utility system.

The utility systems, in turn, generally include the following subsystems:

- Hydraulic system,
- Fuel system,
- Environmental control system (ECS),
- Secondary power system (SPS),
- Landing gear,
- Braking system (Anti-skid),
- Fire detection/extinguishing,
- Life support,
- Anti-ice,
- Recording system, crash/structural.

Figure 2-9 is a top-level diagram that shows the major functions defined for the IVMS. The various subsystems are seen to be highly interactive, and the system represents a large number of control loops with a diverse set of signal interfaces. The system also provides the control for the major energy sources in the system.

As a key element for enhanced flight performance in the next generation of air vehicles, the IVMS will provide the necessary computing resources and communication structure for complex control functions. Increased complexity results from the requirement of integrating many functions for improving performance, extending the flight envelope, and decreasing pilot workload.

2.9.1 Architectural Characteristics

VMS is still in its infancy and is expected to continue to evolve. It must be designed to a stringent set of safety requirements and must provide the capability to accommodate advanced hardware element retrofits (to expand functions and add subsystems with minimum impact on

The analysis of IVMS architectures for optimal functional partitioning has to minimize:

- Hardware interconnections,
- Signal interchange and related timing constraints,
- Failure propagation due to erroneous lack of data,
- Modification required for future LRM enhancement and/or additions,
- Workload for fault location and subsequent maintenance action for the system.

A well-designed IVMS architecture should keep the complexity of system interconnections to a minimum with the constraints set by communication links, LRM size, and fault tolerance requirements

The IVMS should be kept as simple as possible. Safety unquestionably suffers whenever unnecessary complexity is introduced. It is a misguided belief that complexity is a way of achieving performance. Any complexity built into the system should be justified by the performance improvement gained. Reliability and maintainability (including testability) are to be given equal priority to flight safety and performance.

IVMSs are I/O intensive with many different types of interface signals. The present VMS architectures are characterized by a high proportion of analog and discrete signals. Electronics supply sensor excitation voltages and provide servo amplifier output drive capability. The first ingredient of reliable electronic systems is a reduced chip count. The challenge is thus to minimize interface hardware and wiring mass either by LSI implementation and/or by intelligent ("smart") sensors/actuators. Although there are definite trends toward distributed processing, it will probably not materialize in the near future for "smart" actuators. The use of smart sensors/actuators must consider that sensors/actuators typically exist in a harsher environment than an electronic bay.

Utilization of advanced VLSI technology, custom monolithic analog microcircuit technology (CMAM), and ASICs, together with innovative approaches to vehicle installation design and methods, are major prerequisites to incorporating the required functionality in the given constraints (such as weight, space, power consumption). Advances in microelectronics, together with the highest possible degree of standardization, will collectively lower aerospace electronic system life-cycle cost.

The environmental condition experienced by the IVMS is determined by its location in the vehicle. The means of fulfilling the structural design requirements for vibration, shock, and so on, are well known and need not be expanded upon here. The ability to operate in absence of a cooling medium for a predefined time (at least 30 min) is required, and this event has to be indicated to the pilot.

The IVMS and the installation must be protected against lightning strikes and must be electromagnetically compatible with other vehicle equipment ("internal compatibility"). The IVMS must also be capable of operation in a substantial external radio frequency/electromagnetic environment.

Development and verification costs escalate very rapidly as a function of the criticality level of a control task. The IVMS can be subdivided into flight-critical function management (degradation or loss of function can/will result in unsafe operation of the vehicle) and less critical utility function management. As the criticality of some functions varies from vehicle to vehicle, it is impossible to identify which functions are in each of the IVMS subdomains. The partitioning of functions with different levels of criticality and the associated segregation of software risk classes is important. The IVMS architecture must be capable of maintaining separate processing classes. The level of effort required to implement and validate partitioning must be consistent with that required by the most critical function to which partitioning applies.

Partitioning can be achieved with a combination of hardware and software techniques. Memory management mechanisms can be used to ensure that the program and data memory contents of software executing in one processing class cannot be affected by any action in any other processing class. Specialized techniques such as those used in robust partitioning must be used to protect CPU and I/O registers, system status, and the timing of other shared hardware resources.

IVMS design should anticipate the needs for long-term evolution of both the system and the system concept. As hardware costs continue to shrink relative to software costs, the reusability and portability of the system concept and the software are going to be increasingly important.

2.9.2 Software

Control functions to be performed by the IVMS in the future are becoming more comprehensive and complicated as task-oriented control and more complicated maneuvering modes are fully exploited.

The current structure of the embedded software for flight-critical systems reflects the critical nature of these applications. The character of computation exhibits a large degree of regularity (with low complexity), where real time aspects and phase lags are of prime importance. The software structure is by design very simple and implies the repetitive execution of sequences of application tasks at fixed execution rates (with a multiple of a basic frequency). Static assignment of tasks to processors is desired. Task scheduling is not interrupt driven by randomly timed events. Interrupt sources are reduced to an absolute minimum and are only allowed for:

- Time triggering of periodic tasks,
- Exception handling.

Exceptions are either fatal (e.g., power failure) or nonfatal. Fatal errors cause the system to shut down as gracefully as possible and do not return. Nonfatal errors attempt to correct the effects of an exception and then return to normal processing. The design of hard-real-time tasks must consider the time consumed by the worst-case combination of exceptions.

The general method for controlling the execution sequence is by use of a time synchronized (cyclic) executive. The sequence of activities to be performed by a processor within a frame (or shorter subframe) is predetermined due to the implementation of a static preplanned scheduling mechanism. The main advantage of this structure is that it significantly reduces the number of system states that must be verified. This is done by eliminating of the uncertainties related to random interruptions of the execution of critical tasks. The resulting system, although relatively easy to implement and validate, is exceedingly inflexible once iteration rates and task selections are made. A critical task may be preempted only if the preemption was preplanned by static scheduling and only if all I/O and intertask communication are controlled by the static schedule. A noncritical task may be preempted at any time. Priority-based preemptive scheduling may be used as long as the critical task preemption restrictions are enforced. Additional research is needed to relax these restrictions while maintaining the capability to prove correctness.

The key principles IVMS software must be based on are simplicity and visibility. These principles have to be manifested within the specification and production process as well as within the products (reports) generated during the different development stages. The term "simplicity" does not mean that the overall system complexity must not proceed beyond some simple level. It means that at each specific stage, relevant information can be surveyed and reviewed by human analysts. To achieve the required quality standard, intensive control of software production procedures is indispensable. This includes clear and thorough definition of requirements, extensive testing and design audits, detailed documentation, and rigorous production and configuration control.

A consistent system development methodology and supporting tools (for requirement specification, design, implementation, maintenance) dealing with the entire life cycle, of which development is only a part, are essential. There is clear evidence that requirement and software errors introduced late in the development process are the most difficult and costly to detect.

Frequently used software functions have a tendency to migrate into hardware, possibly via firmware as an intermediate step. Thus some of today's software issues can be expected to continue to become hardware issues in the future. The software features that are particularly strong candidates for hardware realization in the future include voting/monitoring, distributed control features for synchronization, intercommunication, and scheduling.

Scope control and data encapsulation, essential in modern programming languages such as Ada, could be provided in hardware with better addressing mechanisms that integrate advanced approaches to protection, eliminating the so-called semantic gap of the von Neumann machine. Objects and capabilities, along with modularity, will contribute to reliability due to a fine grain of protection and fault containment.

2.9.3 Vehicle Health Monitoring and Diagnostics

The IVMS built-in test (BIT) objective is to detect failures as they arise during flight or prior to flight during the preflight check, and to ensure that the system is operating within defined performance limits at all times. Because redundant systems lend themselves to BIT by their very nature, a concept of maximum coverage has to be developed not only with flight operations, but also with test and maintenance activities. The maximum coverage concept has to be considered from the start of the development. Diagnostics and health monitoring in the IVMS should maximize the use of existing redundancy mechanisms to minimize duplication of effort. According to the operational mode of the vehicle, the IVMS will be tested by:

- Continuous built-in test (CBIT),
- Initiated built-in test (IBIT).

Various IBIT functions will be designed in an integrated manner but will be executed in accordance with the IVMS maintenance/servicing concept. However, each level may serve as a maintenance test, depending on the nature of maintenance action required.

IVMS CBIT should provide failure detection, isolation, and reporting by a combination of cross-lane and in-lane monitoring. Trend monitoring (preventive maintenance) should be provided to determine those signal failures that are transient in nature and thus may fail positively in the future.

The IVMS health monitoring system should minimize false alarms, maximize fault isolation, and identify intermittent failures. BIT failure information gained from both CBIT and IBIT should be recorded, and the diagnostic system should make the faulty item (LRU, LRM) visible to the applicable maintenance organization.

In today's three-level maintenance concept (flight line, intermediate shop, depot/factory) with removal/replacement of LRUs at the flight line, the need for the intermediate shop level can be excluded when modules become line-replaceable modules (LRMs). The LRM concept, with a semipermanent installed integration rack, may allow a reduction in spares inventories.

The IVMS has to provide the necessary test interfaces for support equipment to give assistance throughout a project life cycle (i.e., IVMS development, integration, vehicle integration, structural coupling, EMC and production vehicle testing).

2.9.3.1 Artificial Intelligence for Flight Control Maintenance Diagnostics

One of the first uses of artificial intelligence in aerospace was the creation of a maintenance diagnostic tool for aircraft avionics. Using model-based expert systems technology, off-board techniques have been demonstrated for military application on the F-16 Flight Controls Maintenance and Diagnostics System (FCMDS) sponsored by Wright Laboratory, Flight Dynamics Laboratory. This demonstration led to the adoption and implementation of the B777 Central Maintenance Computer (CMC) model-based diagnostic techniques.

The FCMDS program demonstrated:

- A 43% reduction in spares;
- A 26% diagnostic time savings;
- A 92% increase in diagnostics accuracy;
- Zero false LRU pulls;
- An estimated reduction of 2748 maintenance person-hours per month for flight control LRUs alone in the U.S. F-16 fleet;
- Decreased training time.

Onboard and off-board maintenance diagnostics capabilities enable two-level maintenance concepts.

2.9.3.2 Onboard AI Based-Diagnostics

This FCMDS experience led to the concept of embedding diagnostics onboard a new commercial transport aircraft design.

The B777 was designed with the goal of no unscheduled maintenance to provide a significant saving in operating costs. At a minimum, the aircraft was designed to operate for an additional 10 days with a dispatch probability of 99% after the first failure. The B777 CMC, aircraft monitoring functions, and modular avionics architecture support identification and isolation of system failures. High fault tolerance, high reliability, and modest redundancy provide the ability to dispatch with avionics system failures. The system's embedded built-in test and model-based diagnostics techniques facilitate rapid detection and isolation of hardware failures, which reduces personnel support and increases avionic system availability.

The CMC utilizes a model-based diagnostic technique based on accurate computer models of the various aircraft systems. The model is derived from engineering databases that provide close tracking to system revisions; this implies ready-to-deploy diagnostics when the system is fielded. No learning time is required, as is the case with rule-based diagnostics. Furthermore, the development, integration, and checkout of the B777 system were greatly facilitated by the integrated maintenance and diagnostics system. It proved to be a valuable tool in performing subsystem- and system-level engineering tests during development and integration. Confidence in the system performance due to these capabilities in part led to the aggressive first flight demonstration and shortened certification time.

Overall operating cost savings are projected to be in the range of 30%–40% lower than even the recent B747-400. These cost savings amortized over the 30-year life of the aircraft are a very significant value.

2.9.4 Restrictions

The IVMS will have to meet stringent of flight safety and availability specifications. Conservative design practices in engineering impose certain restrictions on the application of advanced hardware/software and microelectronic technologies. Vehicle system components (sensors, computers, actuators, etc.) typically reflect technologies that have been proven in at least a laboratory environment approximately 4 to 5 years prior to first flight.

While VLSI technology offers numerous opportunities, it also introduces several new problems for the IVMS that must be considered. The clearance of commercial standard microprocessors and other complex electronic circuits give rise to the following concerns:

- Commercial definition specification, which is often ambiguous;
- Difference between hardware and documentation;
- Subject to continuing development (e.g., mask sets are updated) without the manufacturer informing the user;
- Changes added by second source manufacturer;
- Transient faults will be more prevalent because extremely low circuit energy level will make devices much more susceptible to external interferences;
- Too complex for totally rigorous test in all possible operating conditions.

New commercial microprocessors use techniques (such as branch target prediction caches and speculative evolution) that make average execution times faster but make worst-case execution times slower.

Theoretically, neither simulation nor testing can completely verify the correctness of a VLSI circuit. As VLSI designs increase, the issue of correctness grows from an interesting question into a practical consideration with which equipment supplier and airframe manufacturer must deal. One cannot rule out the possibility of residual hardware design faults at the chip level. Of grave concern is the possibility that one or more very rarely entered processor states might represent a hazardous generic design flaw, and that such a state might be entered into (essentially simultaneously) by all processors in all redundant lanes. The same problem can show up with ASICs, where the equipment supplier is responsible for the testing procedure and no high-volume customers can assist in finding errors. Current trends will increase the problem because VLSI circuits are just as complex as software.

The methods for alleviating type acceptance problems are strict configuration control to VLSI circuit mask level, traceability of each batch of components used, and robust design of the computer architecture to detect/absorb unexpected VLSI behavior.

The integrity of the IVMS software has to be compatible with the given integrity requirement for the system and is regarded as Risk Class 1 for safety critical control functions.

The absence of credible reliability prediction methods (Risk Class 1) for software to be used in safety-critical applications will impose problems for the IVMS reliability analysis. The length of time under test and the small samples of failure observation make it impractical to assess the reliability of the software by means of growth and statistical models.

In practice, no matter how carefully the software is designed, it is impossible to establish that it is completely error free because the large number of possible states precludes exhaustive testing. Furthermore, the statistical analysis methods used for random hardware failures are not applicable to software or hardware design.

A "proof-of-correctness" method for validating software was investigated as part of the SIFT program. However, the SIFT proof of correctness only covers the operating system and not the application domain. This technique is also not readily understandable by the engineering community.

Formal specifications with proof of correctness for IVMS are still years away.

Since we know of no satisfactory way to estimate the probability that a software load is incorrect, we are forced to guarantee that the software is indeed correct. To address the safety aspects and provide the necessary evidence that leads to flight certification, the following approaches are currently followed:

- Subsets of a high-order language (HOL) are used for all safety critical software;
- Verifying the HOL source code using static analysis tools;
- Verifying the integrity of the compilation process by analysis of the object code produced.

Furthermore, software safety is achieved not only by the application of analysis techniques but also by providing safety-related requirements/guidelines for the software design, coding, and testing. The compliance of the software design/code with the guidelines has to be cross-checked.

Major control system developments for safety-critical applications that have been certified in the last decade and that have brought this technology to a highly mature state typically contain 100-200 Kbyte of Risk Class 1 software. Significant improvements in applying software engineering principles are necessary to cope with the associated safety issues when the size of the IVMS software exceeds more than 1 Mbyte. Mature tools for system analysis, software requirements analysis, and software development and maintenance are essential. They allow requirements to be analyzed for completeness and consistency and provide support for development phase independent activities (e.g., configuration management, project management, quality assurance, documentation).

Verification and validation of the IVMS cannot be considered an afterthought of the development process but must appropriately influence the entire process from the very beginning. Verification (as the process of showing that the software fulfills its specification) can only increase the confidence level that all hardware/software errors have been detected and eliminated. The validation process must demonstrate that the specifications are complete and consistent, and that the IVMS meets its requirements and the vehicle reacts correctly in all situations.

2.10 FAULT TOLERANCE DEFINITIONS

The purpose of this section is to present a set of definitions for fault tolerance terms. Most of the terms defined here have developed some diversity of meaning. This diversity typically causes ambiguity and confusion in fault handling discussions. The definitions presented are a "best" compromise between wide acceptance, consistency, and usefulness.

The basic definitions fall into three categories: (1) impairments to device functionality, (2) means of mitigating these impairments, and (3) measures of device characteristics influenced by faults.

2.10.1 Impairments

The first four definitions, failure, fault, error, and latent, are terms related to impairments to device functionality. These definitions are essentially the same as those adopted by the Fundamental Concepts and Terminology subcommittee that is common to the IFIP Working Group 10.4 on Reliable Computing and Fault Tolerance, and the IEEE Computing Society Technical Committee on Fault Tolerant Computing [2.3].

Failure—The deviation of a device's service from the expected service proscribed by some agreed-to specification. The deviation, and therefore the failure, can be classified by type, persistence, and degree of severity. A measure of the degree to which the failure may adversely affect the device's service is called the criticality of the failure.

Error—The part of a device's state that differs from the state that would exist in an unimpaired device. The term *state* usually refers to information stored in a device. However, the definition of this term sometimes is stretched to include the structure of a device as well.

Fault—The phenomenological cause of a failure. When the stretched definition of state is used, a fault is the cause of an error in the device's structure. Failures and errors are effects of a fault.

Latent—An adjective used to describe an impairment that has occurred in a phenomenological sense but has not yet affected the device's operation. It is most commonly used in the phrase "latent fault." When the stretched definition of state is used, the phrase becomes "latent error." Note the difference between *latent* and *incipient*. The latter means a predisposition to occur or in the process of occurring, whereas the former means an actual occurrence that has not yet produced any effects.

Because of the recursive definition of *device* (an aggregate device can be composed of a number of component devices), the above impairment definitions have a recursive quality to them. For example, a fault in a component device can cause errors in or a failure of that device. The failure of the component device, in turn, can cause errors in or failures of the aggregate device. This can proceed through any number of levels in an arbitrarily nested hierarchy of devices.

The converse of the preceding paragraph is also true. An examination of a faulty device may show that the fault can be attributed to a component of that device. Further examination may find the origin of the fault to be a device within the component, and so on. It can be seen from this discussion that the basic cause of a particular fault may depend on the capability of available fault analysis to be precise, and it also may be a judgmental issue [2.4].

The preceding discussion dealt with the precision with which a fault can be described and not with fault propagation. Fault propagation is the inducement of other faults, failures, and errors in additional devices. The additionally affected devices may be related to the original faulty device in a hierarchical fashion or they may be peer devices of the original faulty device. The induced faults, failures, and errors may persist after the original fault has been corrected.

A Concrete Example

As an example, consider a stuck-at-one fault in the lowest bit of a RAM memory word. The fault is the flip-flop's inability to switch to the zero state when required. The service that the RAM word should provide is storage of an arbitrary bit pattern. As long as only odd numbers (lowest bit equal to one) are stored in this word, the fault remains latent. When the RAM word is used to store an even number (lowest bit equal to zero), the fault becomes active and a failure occurs. (The RAM does not perform its specified service of storing a bit pattern.) The error corresponding to this failure is the lowest bit being a one instead of a zero. This is a latent error until the word is read.

Assume that this memory word is part of a processor that is to compute some function involving the information in that word. The error in the memory word could cause the function to be computed incorrectly, thus causing a failure of the processor. With low precision of fault analysis, the basic fault could be judged as a processor fault.

An examination to determine the cause of the flip-flop's inability to switch to the zero state might uncover a short in a transistor. This, then, would be judged the fault. If it were determined that the short was caused by an interruption of the E-beam during chip creation, that would be judged the fault.

An Abstract Example

Any digital system can be represented by a finite-state machine. The impairments to the digital system can also be represented in terms of the state machine. The service a state machine provides is the mapping of a current state and an input to a specified successor state. A failure is represented by the state machine mapping a given state and input to a state that violates the specification. The associated fault is the act of transitioning from the current state to the incorrect successor state. The associated error is the difference between the specified successor state and the state reached because of the fault.

2.10.2 Means

Many means of coping with the possibility of having faults exist Below are definitions used to describe some of these means.

Fault Avoidance—One way of coping with the possibility of having faults is to design and build devices that have a low probability of failure. This generally means the use of conservative and "overengineered" designs and construction techniques. However, as the requirements for the probability of failure become more stringent, the cost of fault avoidance very quickly becomes prohibitive. (This includes opportunity costs such as loss of throughput.) In many applications, it is not even technologically feasible to meet the requirements using fault avoidance alone.

Fault Tolerance—After exhausting the reasonable fault avoidance possibilities, the only recourse for handling the remaining faults is through fault tolerance. The description of the fault tolerance for a particular device must include the specific faults or fault classes that are tolerated. For each such fault or fault class, the description must specify a post-fault level of service. It is a meaningless statement to say that a device is fault tolerant without explicitly stating these specifications. (All devices are fault tolerant with respect to some class of innocuous faults, and no device can tolerate all possible faults.)

It is important to realize that fault tolerance is a means and not a measure. The question: "How fault tolerant is the device?" is not directly answerable. No measure can be given to answer the question. The question can only be answered by explaining how the fault tolerance affects the measures defined below.

The description of fault avoidance stated that fault avoidance by itself is too expensive to be the sole means of handling the possibility of a fault. The converse is also true; a system cannot afford the resources required if fault tolerance alone has to cope with all possible faults. A fault-tolerance strategy must fit into a higher level fault-handling strategy that ensures the effectiveness of the system. This strategy must be composed of a complete spanning set of compatible strategies for fault avoidance, fault tolerance, and high-performance design.

The process of tolerating a fault is usually described as occurring in three steps: fault detection, isolation, and recovery.

Fault Detection—This term continues in general usage even though it is a slight misnomer. In practice, faults are not directly detected. Instead, fault detection methods are actually failure or error detection methods. The existence of a fault is inferred from the existence of these effects of a fault. There are numerous schemes for detecting faults in a large number of fault classes; however, none of them is perfect. The measure of this imperfection is described by coverage (see definition below).

Fault Isolation—After the effects of a fault have been detected, a more in-depth analysis is performed to determine the type and location of the fault and to determine which components of the device may have been adversely affected by the fault. The depth of this analysis need only be to the level required to make correct decisions during fault recovery.

Fault Recovery—The steps of fault recovery include: fault containment, which are means to limit the effects of a fault; the fixing (actual repair of the fault) or masking (inhibition of the fault's effects) of a faulty component such that the fault no longer affects the device's operation; and the correction of any errors. The result of fault recovery is the provision of some degree of post-fault service as specified by the device's fault tolerance requirements. The degree of post-fault service can be partitioned into many classes. The three most common classes are fail-op, graceful degradation, and fail-safe. These terms are described below as measures of a device's service.

2.11.3 Measures

Because fault tolerance is a means to an end, the end must be well defined before specific means can be determined. The intent of fault tolerance is to give a device a set of characteristics. The most notable of these are reliability, availability, safety, and security. The reason it is important to differentiate these characteristics is that the fault tolerance techniques that are appropriate for achieving one characteristic may be a hindrance to achieving another characteristic! For example, one common technique for achieving high availability is to create pools of redundant components tied together by shared communication channels. This technique, however, is in direct opposition to the isolation techniques used for safety and security.

Reliability—Reliability is defined by a mathematical function, R(t), which evaluates to the probability that a device will have provided acceptable service from some initial time until time = t. It has been generally assumed that electronic components have a hazard rate that follows the shape of a "bathtub curve." This assumption has not been proven for VLSI. If the assumption is true, then a VLSI device would, for most of its life, have a constant hazard rate. If the hazard rate is constant, then $R(t) = e^{-\lambda t}$ where λ is the device's failure rate (which is also the inverse of mean time between failure, MTBF, when the hazard rate is constant).

Availability—For devices that are fault tolerant or are repairable, availability is the fraction of time that the device can provide some minimally acceptable service. Generally, availability is a function of mean time between failures (MTBF) and mean time to repair (or mean time to recover for fault-tolerant devices) (MTTR). Availability is not concerned about the quality of the service beyond the minimal acceptable service. Point availability, interval availability, and steady-state availability each are measures of availability with respect to different time intervals.

Integrity—This is the measure of a device's ability to produce specified quality of service. It is the probability that the device will not produce incorrect service. Integrity is not concerned with the possible absence of service, just the quality of service if produced. This shows that integrity and availability are complementary measures. Two major types of integrity are safety and security.

Safety—This is a measure of a device's ability to provide a quality of service that does not pose a danger to the existence of human life or the existence of high-value assets (such as an aircraft). This is usually measured as the probability of a catastrophe for a given exposure (typically one hour, one flight, or one mission).

Security—Security is the ability of a device to provide a service only to authorized recipients of the service. This is becoming a great concern among the designers of future integrated systems that may have to handle multiple levels of classified information.

There are two concerns with respect to security. One is that a component malfunction will cause a breach of some security constraint. The second is the possibility of propagation and emanation of spurious or parasitic classified signals. This has led to the TEMPEST Red/Black criteria, which state that all classified and classified related signals are considered Red and all signals that are available to the enemy are considered Black and never the twain shall meet without encryption and special filtering interfaces, including power supplies and test connections.

The consequences of security failures can be grouped into input or output problems. The former is sometimes called spoofing or imitative deception. This allows an enemy to gain access to the system. This access may allow the enemy to insert incorrect data or to "take control" of the system and subvert its purpose. The second consequence group, output problems, is sometimes referred to as security leaks. The adverse effects are obvious. For example, if the message from a commander to a weapon is leaked to the enemy, the enemy can take action to protect the intended target before the weapon is effective.

Survivability—The ability of a device to withstand the hazards created by any external hostile action. It is usually defined as the probability that the device will not suffer service degradation below a specified limit within a specified time period. This time period is usually one mission time, but sometimes it is specified on a per-hour basis.

Dependability—Dependability is the aggregate term for reliability, availability, integrity (safety and security), and survivability. It is normally used only in a qualitative sense because the component measures are so diverse. To be used in a quantitative sense, dependability would have to include the numeric values for each of its component measures.

The most important factor in determining the worth of fault tolerance is the quality of service that can be provided after a fault has occurred. The three most common classes of post-fault service are described below.

Fail-Op—This is the ability of a device to remain fully operational (provide the full specified service) after the occurrence of a fault. This is usually called fail-operational and is abbreviated to fail-op. It is common to use the mathematical notation of powers to indicate the number of independent failures that can occur while the device still maintains full service. For example, fail-op2 (read as "fail-op squared") means the device provides full service even after the occurrence of two independent faults.

For this fail-opN notation to be a meaningful measure, a list of the independent faults to be tolerated and a measure of the independence must accompany it. This is seldom done in practice, so statements about fail-operability are usually meaningless. Sometimes this notation is used with respect to an implied set of common benign faults (stuck-at-zero/stuck-at-one, open, and short).

Graceful Degradation—If a device cannot provide full service after a fault, it may still may provide some degree of service. If the degree of service gradually decreases with the occurrence of faults, the device is said to have graceful degradation. Sometimes this is called fail-soft. For such a device, some measures of the degree of service performed under various faulty conditions must exist. One such measure is called performability. [2.5].

Fail-Safe—If a device cannot provide any service after a fault, a question arises about the safety of this no-service condition. A device that need not provide any service to remain safe is called a fail-safe device. This term is often combined with fail-op, such as: fail-op/fail-safe. This example means that the device can remain fully operational after one fault and remain safe after a second fault. The same meaningfulness caveat applies here as it does for the fail-op measure (e.g., the class of tolerated faults must be stated). In addition, the set of safe conditions must be defined.

Coverage—The ability of the fault tolerance mechanisms of a device to detect faults is expressed as a fraction, called coverage. This fraction can represent the probability that a particular fault will be detected, it can represent the fraction of all faults in a fault class that can be detected, or it can be a combination of both. Coverage is a measure of the ability to correctly signal that a fault has occurred when a fault has indeed occurred. Another possible imperfection in fault detection is the indication that a fault has occurred when, in fact, no fault has occurred in the device (except the fault in the fault detection mechanism that is indicating some other fault exists). This is called the false alarm problem. There is no common named measure for false alarms.

Another imperfection in coverage is the time between the occurrence of a fault and the time it is detected. Because of the close link between this imperfection and the idea of a latent fault, this is usually called fault detection latency or coverage latency.

2.10.4 Dependability Implications for Design

As the requirements for dependability increase, the cost of a device increases by requiring more reliable components, more redundancy, more thorough validation and verification (V&V), etc. In the region of dependability, with which we will probably be concerned, the relationship is not linear. The increase in device cost will probably be exponential with respect to an increase in dependability requirements. It is therefore important to be as precise as possible in determining the component measures for dependability.

The relative stringency of requirements for the "ility" measures is also important. This balance can have a great influence on the fault tolerance design. For example, a typical fault detection technique is to have a pair of redundant components, say component A and component B, that produce outputs that are compared. An error is indicated if the outputs are significantly different. In software, the comparison would look something like:

```
IF |outputA - output B| > e THEN failure occurred
```

The designer is faced with a tradeoff for this design. This tradeoff is the selection of e. If e is made too large, some errors will be undetected. If e is made too small, there will be too many false alarms. To make the correct selection of e, the designer must know what are the goals of the fault tolerance design. If the goal is high availability, a large e should be chosen. If the goal is high integrity, a small e should be chosen. Tradeoffs that depend on the relative stringency of requirements abound in fault tolerance design.

2.11 ROBUST PARTITIONING

Physical integration increases the risk that unwanted interactions among the functions residing on the shared hardware will lead to unforeseen failures. It is no longer sufficient to write an ICD that defines explicit interactions occurring over separate communication paths. If a physically integrated architecture is to be implemented successfully, *all* possible adverse interactions (explicit, implicit, or unintended) must be controlled. These problems come from several sources:

- There is an increased number of possible fault propagation mechanisms between "independent" functions. And the number of possible fault propagation paths for each mechanism grows with the square of the number of functions integrated.
- Multiple subsystems, which are inherently but not explicitly redundant, can be simultaneously lost if they depend on the same resource. An example of inherently redundant subsystems is altitude from radar altimeter, baro altimeter, INS, GPS, etc.
- Functions of different criticalities can share the same resource. In this case, the higher criticality function(s) must be protected from the lower criticality function(s). (To be affordable, lower criticality functions are usually developed with less rigor than higher criticality functions and are therefore less dependable.)

The problem of unintended adverse interactions between subsystems is the main unique problem with physically integrating a VMS, which is not significant in a nonintegrated VMS nor in an integrated system that is not flight critical.

The best approach to this problem is to attempt to make the execution environment of each function in the shared hardware as much like the environment in the discrete LRU as possible. Essentially, all resources of the shared hardware must be rigidly partitioned to ensure that one function cannot adversely affect another under any possible operating condition, including the occurrence of faults.

Functions must be deterministically partitioned both in space and in time; that is, it must be determinable at design time that no function can adversely effect another function's space or time partition as defined here. Space partitioning refers to the uniquely allocated resources of a function. These resources are allocated to the function at design time from the pool of available shared resources. Thereafter, these resources are wholly controlled by that one function, even though these resources may reside within the larger shared resource pool. Time partitioning refers to the time allocation of a resource that cannot be space partitioned. Thus resources are usually processor CPUs and data buses.

Deterministic control over the partitioning of space means that it can be guaranteed that no function can prevent another from obtaining adequate memory space and that the memory space assigned to one function cannot be corrupted by the behavior of another function. Preallocated memory areas prevent contention for memory space. Hardware-based memory protection mechanisms, such as processor memory management units (MMUs), are usually adequate to prevent corruption on uniprocessors. For multiple processors, which share multiport memories, a simple MMU is insufficient. Other techniques, such as those described in Subsection 6.9, are needed. Non-shared I/O and other special-purpose registers are the other major resource that can be space partitioned.

Deterministic control over the partitioning of time means that it can be guaranteed that one function's variable demand for shared hardware resources will never prevent another function from obtaining a specified minimum level of service and, more importantly, that the timing of a function's access to these resources will not be affected by variable demand or by the failure of another function. If the system design does not build in time determinism, a function can be verified only after all possible combinations of events, including all possible combinations of failures, have been considered. Clearly this would drastically increase the cost of verification, as well as of software maintenance.

2.12 CONCLUSIONS AND RECOMMENDATIONS

The known methodologies to achieve integration (i.e., physical and functional) have been described. It was shown that physical and functional integration have unique advantages, namely:

- Physical integration: provides hardware reduction, then cost reduction.
- Functional integration: provides performance optimization and crew workload reduction.

Clearly functional and physical integration are different but also complementary, and they could be joined to utilize the benefits of both. A comparison between functional and physical integration was made to show their peculiarities and differences using some examples. Such analysis starts with the assumption that both methodologies (physical or functional integration) are useful in solving integration problems and can result in substantial differences in the resulting system. The choice of the proper methodology (physical or functional integration) or of a mix of them should be performed on a design requirements basis. Specifically, performance, cost, safety, and integrity requirements should be adequately taken into account during this evaluation because their relative weight plays a relevant role in the choice or in the balancing of integration methodologies.

Although system complexity increases due to the introduction of new functions, integration should became more and more important in system development. It can provide cost reduction and crew workload optimization, and can positively influence system reliability and maintainability. In addition, it is an exciting feature as system reconfiguration becomes feasible and the introduction of other complex functions is simplified with respect to non-integrated systems.

System integration can also produce failure tolerance and integrity benefits, contributing to spares reduction and the graceful degradation of a system after a failure. On the other hand, the introduction of new functions can require the adoption of new technologies that can increase the technology risk. The necessity to maintain the technology risk at an acceptable level requires a case-by-case evaluation. The functionalities forming the system have to be analyzed and grouped taking into account their coupling (the necessity to perform extensive data exchange), the separation between safety and non-safety-critical functions, and other design constraints. Integration is aided by such actions, and its effectiveness is increased. In this way, the technology risk can be kept at an acceptable level.

Inside the family of integrated systems, IVMS represents a particular case leading to intrinsic constraints mainly related to failure tolerance, integrity and safety critical requirements. In particular, integration of functions that have differing fault tolerance, integrity and safety requirements force new architecture design techniques such as robust partitioning. Therefore, IVMS will have more severe and constraining requirements than other avionic systems. This leads to the necessity to keep it as simple as possible, so that the introduction of additional complexities must be justified by suitable gains in performance. This requirement also leads to a separation between functions based on their criticality level, enabling the separation of software risk classes. Integration between software risk classes should be kept under control to avoid conflicts between software classes of different criticalities.

2.13 REFERENCES

[2.1] Blackman, S. "Multiple Target Tracking with Radar Applications," Artech House Inc., 1986.

[2.2] Avalle, M. "An Integrated System for Air to Ground Operations," AGARD 2nd MSP Symposium, Rome, Italy, October 1994.

[2.3] J. C. Laprie "Dependable Computing and Fault Tolerance: Concepts and Terminology," *Proceedings, 15th Fault Tolerant Computing Symposium*, Ann Arbor, MI, June 1985; IFIP WG 10.4 Summer 1984 meeting, Kissimimee, FL; and LAAS Report No. 84.035, June 1984.

[2.4] D. Morgan, W. C. Carter, and A. Hopkins, "Report to IFIP WG 10.4, Concepts and Terminology, Draft 1," IFIP WG 10.4 meeting, Como, Italy, June 1983.

[2.5] J. F. Meyer, "On Evaluating the Performability of Degradable Computing Systems," *Proceedings, 1978 Fault Tolerant Computing Symposium*, Toulouse, France, pp. 117-143.

[2.6] McDonnell Aircraft Company, "STOL/Maneuver Technology Demonstrator, Vol. 1, Executive Summary," Flight Dynamics Directorate, Wright Laboratory, WL-TR-91-3080, September 1991.

CHAPTER 3

ASSESSMENT OF INTEGRATED VEHICLE MANAGEMENT SYSTEMS (IVMS) DESIGN

3.1 INTRODUCTION

The enhancement of vehicle performance or an increase in vehicle capabilities are general reasons for integrating vehicle functions into an IVMS. However, the increased benefits do not always come free and are often offset by certain associated penalties brought on by the resulting integrated system. The purpose of this chapter is to explore both the benefits and penalties associated with the development and deployment of an IVMS and to provide a framework for assessing IVMS designs. As a specific IVMS design solution is dependent on a particular vehicle type and mission, it is difficult to discuss specific benefits and/or penalties of an IVMS in general system design terms. Thus, a generic approach will be taken so that the benefits and penalties can be applied to a broad category of IVMS applications. Some examples of vehicle systems that can be considered part of an IVMS are discussed in this chapter, including the associated benefits and penalties of their design.

3.2 VEHICLE REQUIREMENTS AFFECTING IVMS DESIGN REQUIREMENTS

The IVMS approach for integrating vehicle subsystems is viewed as an essential element for meeting stringent affordability and performance requirements for future aircraft. Other factors of importance are availability and survivability, and again the IVMS appears to offer improvements for these factors as well. In this era of defense downsizing, affordability is a prime factor that must be considered because fewer aircraft are being purchased. Enhanced performance is always a desired factor, and the ability to extract more vehicle performance through integration of subsystems can provide the winning edge. With the decreased numbers of aircraft being procured, availability is important to offset decreasing aircraft inventories. Downsizing means not only fewer aircraft but also proportionally fewer maintenance personnel and facilities to do the job, and the issue of logistical support continues to be important. Finally, fewer aircraft also means that the aircraft must be survivable under all conditions, such as adverse environmental and EMI/EMP threats and foreign object impact conditions.

As discussed in Chapter 1, IVMS integration is achieved at both the physical and functional levels. Physical integration can be achieved though the use of a data bus that allows the coupling of several functions; the use of load sharing, such as a processor performing two or more separate tasks; sensor sharing, as in the case of a common inertial measurement unit for both the flight control and navigation functions; and the use of common module building blocks across all IVMS functions. Functional integration is attained through the coupling of two or more functions to achieve a higher level of vehicle performance or improvements in other parameters of interest. Examples of functional integration are flight and propulsion control and thermal management, which might couple environmental control, fuel management, and electrical power utilization. Although physical integration can be achieved without functional integration, the opposite is not true, at least from the standpoint of doing it efficiently. Physical integration can be considered an enabling technology for functional integration.

The benefits of IVMS are achieved through either or both functional and physical integration. Table 3-1 illustrates that physical integration can provide benefits in affordability, survivability, and logistics, and functional integration can provide benefits in performance survivability and logistics.

Table 3-1. Benefits of Physical and Functional Integration

Types of Integration	Performance	Survivability	Affordability	Logistics
Physical		X	X	X
Functional	X	X		X

While it is generally believed, and usually can be proven, that an IVMS will provide benefits in performance, affordability, availability (logistics), and survivability, it is also known that certain IVMS designs may have cost and performance penalties as well. The relationships of possible benefits and penalties associated with an IVMS design are shown in Table 3-2. Also shown are the associated penalties that may occur and the metrics used to quantify the benefits and penalties. These relationships are discussed below for each of the factors.

Table 3-2. Assessment of Integrated Vehicle Management System Design

			Assessment Factors		
Vehicle Design Requirements	IVMS Design Requirements	IVMS Enabling Designs	Metrics	Benefits	Penalties
Affordability	Development Cost	• Sharing of functions by components of system • Common modules • Integrated design environment	• Cost of components per aircraft • Interoperability between aircraft • Weight	• Fewer Quantity of LRUs • Fewer quantity of LRU's components • Cost to upgrade • Shared Control/Diagnostic information	• Common modules may weigh more or be unnecessarily complex for a given application • Investment of for development of tools and modular development • Complex testability
	O&M Cost	• Automated diagnostics • Fault-tolerant design	• Development time/cost • Field mean time to repair and field spares inventory	• Lower maintenance cost • Reduced Spares • Lower field Maintenance time • Extensibility	• Increased development cost • Increased V&V cost
Logistics	Operation Availability	• Fault-tolerant/ highly reliable system/onboard "pooled" spares	• Mean time between maintenance actions • Mean time to repair • Number of sorties without maintenance	• Multiple sorties without maintenance • Rapid Response • Maintained force size • Fewer aircraft required	• Increased system cost and/or weight
	Supportability	• Onboard self-test/repair • Modular design • Onboard spares	• Field spares • Maintenance personnel reqm'ts • Reduce levels of maintenance	• Simplified maintenance in field • Reconfiguration	• Increase in development cost • Increased weight • Revalidation when system is modified
	Maintainability	• Automated diagnostics • Onboard "pooled" spares • Common modules	• Mean time to repair • Field time in maintenance	• Increased availability • lower O&M costs	• Increased system cost and/or weight
Survivability	Flight Safety	• Redundancy management • Highly reliable fault-tolerant design	• Number of aircraft losses per flight time	• Increased flight safety • Low probability of loss of control	• MTBF (Maintenance) cost • Increased weight and V&V cost • Increased failure modes
	Vulnerability to Foreign Object Damage	• Fault-tolerant/self-healing configuration • Distributed redundant design	• Battle/foreign object impact tolerance	• Increased probability to survive foreign object impact	• Possible increase in MTTR • Increased V&V cost
	Vulnerability to EMI/EMP Interruption/ Damage	• Shielding/circuit protection design • Fiber optics	• EMI/EMP protection level • EMI/EMP shielding weight	• Survivable in EMI/EMP environment	• Increased weight • Increased MTTR • Development cost of fiber optics
	Susceptibility to Collision	• Integrated crew station with automated collision avoidance systems • Pilot information fusion	• Crew workload; foreign object awareness	• Reduced probability of collision with ground and air objects	• Increased V&V cost • Observability (emissions)

Table 3-2. Assessment of Integrated Vehicle Management System Design (concluded)

			Assessment Factors		
Vehicle Design Requirements	IVMS Design Requirements	IVMS Enabling Designs	Metrics	Benefits	Penalties
Performance	Flight Envelope Capability	• Integrated propulsion and flight control • New control modes	• Flight envelope • Reduced crew section	• Fly closer to control limits/reduce margins • Control tailoring	• Possible reduced mission safety margin • Cost
	Subsystem Support Optimization	• Integrated subsystem power and cooling control • Performance-seeking control	• TOGW • Thrust-specific fuel consumption	• Reduced weight, power, and cooling requirements • Energy management	• Increase criticality of previously less critical systems
	Mission Effectiveness	• Integrated flight guidance, navigation, and mission control	• Kill ratio • Energy management -Fuel -Range • Crew workload	• Increased ability of pilot to respond to mission, flight issues	• Increased weight • S/W complexity
	Pilot Situation Awareness	• Pilot information, data fusion	• Mission effectiveness • Kill ratio • Attrition rate • Crew workload	• Increased ability of pilot to respond to mission, flight issues	• Increased V&V cost
	Coordinated Multiship Tactics	• Intership data integration	• Target kill per sortie	• Minimize combat losses • Maximize capability • Increased weapon effectiveness	• Increased V&V cost • Increased cost

3.2.1 Affordability

Affordability is a factor that requires special consideration and should be looked at in terms of life-cycle cost rather than developmental costs only. A properly conceived and designed IVMS using the principles of physical and functional integration should result in a system that has a significantly lower life-cycle cost (as compared to a traditional system designed with similar performance characteristics), but in fact may need more resources to develop. Lower life-cycle costs can be realized in physically integrated systems if automated diagnostic systems and fault tolerance features are incorporated into the design. These technologies can usually be incorporated into physically integrated systems without a large corresponding increase in system complexity. Onboard, automated diagnostic systems can lower maintenance costs by reducing maintenance time required in the field. The use of common modules will reduce both maintenance time and spares inventory costs in the field. Although common modules may simplify the system design process, these savings may be offset by the cost incurred if no applicable, off-the-shelf common modules are available. Therefore, new common modules must be designed and developed for the particular IVMS application.

3.2.2 Logistics

The key drivers that determine the cost of logistics of an aircraft system are the costs to support the aircraft and the costs to achieve the desired or required level of operational availability. Since IVMS designs are inherently physically integrated, incorporation of common modules that are easily identifiable and replaceable can greatly improve the supportability of an aircraft system by simplifying maintenance demands on support personnel. Common modules will also reduce the amount of spares needed, which will lead to operational savings as well.

IVMS designs can improve operational availability by using the physical and functional integration concepts of onboard pooled spares. This technology has gained acceptance in systems such as the F-22, which was conceived and developed by the USAF PAVE PILLAR program. In this concept, the processing or functional load is shared over a set of common "pooled" modules. If and when these functional modules fail, the load is automatically redistributed over the remaining modules. In this design, the performance (in terms of response time) gradually decreases if no maintenance occurs, but the aircraft can continue to operate without maintenance until either a specified time passes (scheduled maintenance) or the performance degrades to the point where it becomes necessary to perform maintenance. A system using this design approach can achieve multiple sorties without maintenance and can continue to operate with less unscheduled field maintenance than traditional federated systems. However, the development costs may be higher for IVMS designs using the common module and pooled spares concept due to the increased software development costs. The V&V costs are also likely to be higher due to the increased number of operational variations that must be tested.

3.2.3 Performance

One of the principal benefits of an IVMS design is an increase in system performance. Performance benefits can be realized in the following ways:

- Improved flight envelope capability, where an aircraft can fly closer to the aircraft control limits and controls can be tailored to improve the ability of the aircraft to perform its designated mission. This capability can be achieved by functionally integrating the propulsion, flight, and fuel controls and by adding new, advanced control mode designs.
- Improved pilot situation awareness and mission effectiveness, where the design permits the pilot to effectively respond to mission and aircraft flight issues. This capability can be achieved by simplifying the pilot interface and automating many of the aircraft systems that do not require operator control. Pilot interfaces have been greatly improved over the past years by both physically integrating the displays and controls and functionally integrating the information through data fusion and artificial intelligence technologies.
- Optimized subsystem control, where peak demands of various subsystems, such as environmental and electrical power controls, can be reduced and controlled over the flight regime. The benefits of this approach are the ability to use smaller and therefore lighter weight subsystems in the aircraft design.
- Improved multiship coordinated tactics, where a number of aircraft can operate in a highly coordinated manner during air-to-air combat, air-to-ground attack, and other multiship operations requiring a high degree of ship-to-ship flight path coordination.

The principal methods used to increase aircraft performance in the ways listed above require extensive use of functional integration and data and information fusion. Incorporating these technologies into the aircraft design will require more development effort in terms of algorithm and associated software development, as well as in terms of V&V testing.

3.2.4 Survivability

Survivability is a key design requirement in all aircraft, commercial or military. Design factors contributing to the survivability of the aircraft are flight safety features and how vulnerable the aircraft is to foreign object damage, EMI/EMP, and collision with other aircraft. Because an IVMS is both functionally and physically integrated, the designs can be made very resistant to foreign object damage and collisions and can be made to have superior flight safety features.

Until the past decade or so, flight safety improvements were achieved by using multiple, redundant systems. IVMS designs that use the pooled spares concept (a physically integrated design) or other fault-tolerant designs that use functional integration have greatly increased the flight safety of aircraft. However, the designs must be carefully controlled so that critical flight control functions are not contaminated by other non-flight-critical functions. More V&V testing is usually required in IMVS designs to ensure the integrity of the flight control functions in all phases of flight operations.

Because IVMS designs can be made to be physically distributed, logical placement of IVMS elements an aircraft can be designed to be resistant to foreign object damage and EMI/EMP effects. Also, functional integration of selected crew station data and automated collision avoidance systems have been shown to greatly reduce collision susceptibility.

3.3 VMS EXAMPLES

Integrated control modes couple two or more subsystem control systems to increase vehicle performance and provide a system optimization function. The integrated control modes and data sharing provide the basis for functional integration in the IVMS. Examples of integrated control modes include: integrated flight/propulsion control, dynamic electrical load management, active center of gravity (CG) control, and integrated fuel/thermal management. The types of control modes to be utilized in a particular application will depend on vehicle mission, cost, and performance requirements. Several examples of integrated control modes are described in Subsections 3.3.1 through 3.3.9, below.

3.3.1 Integrated Propulsion and Flight Control

Integration of the flight control and propulsion systems offers several opportunities for improving aircraft operational capability. By taking advantage of the cross-coupling between the aerodynamic control surfaces and the propulsion system elements (engine, inlets, and thrust vectoring nozzles), several performance benefits can be achieved. These benefits include improved aircraft maneuverability, precise flight path control, increased fault tolerance, reduced pilot workload, and reduced observability. Performance improvements are brought about by flight envelope expansion, which results in a more maneuverable aircraft. For low speed, the thrust vectoring becomes more effective than the horizontal tail, resulting in precise path control. Blending of the control surfaces, engine inlet and nozzle can reduce trim drag, resulting in extended range. The proper blending of these control effectors can also reduce the aircraft observability by reducing control surface deflection.

3.3.2 Integrated Flight Control and Navigation

When strapdown navigation became the preferred approach with the advent of solid- state inertial sensors (e.g., ring laser gyros), the concept of sharing rate and acceleration information with the flight control function became a reality. This has important cost benefits in sensors but presents challenges for the sensors and implementation. Some of these are:

- Sensor location of navigator is not ideal for flight control (e. g., accelerometers at the pilot station and rate sensors at anti-nodes and a separated set for survivability).
- Redundant high-cost sensors for flight control fault tolerance.

To overcome some of the high sensor cost issues, the notion of using skewed axis sensors has been developed to create higher levels of operation after failures with the fewest levels of failure. One concept, shown in Figure 3-1, has been built for the Boeing 777 integrated flight control/inertial reference/air data system. This hexad arrangement provides the same fault tolerance (2 fail-operational plus fail-safe) as three triads, which use nine gyros and accelerometers.

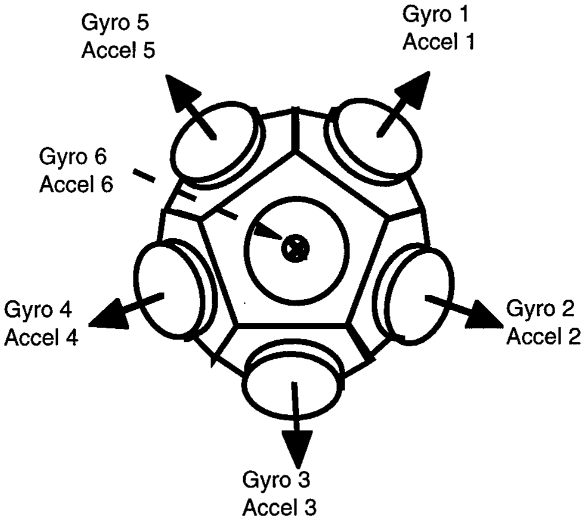


Figure 3-1. Fault Tolerant Sensor Cluster

A similar concept for the military uses two tetrads (four gyros and four accelerometers) for the same fault tolerance, but can be separated for survivability.

3.3.3 Performance-Seeking Control (PSC)

NASA flight demonstrated the integrated airframe/propulsion PSC [3.1] system on an F-15 aircraft. The PSC program integrated several aircraft systems, including the flight control, engine inlet control, engine/nozzle controls, and air data system, as shown in Figure 3-2. The PSC system optimized performance by comparing actual performance with models that are continuously updated in the Kalman filter. The model-based control algorithm adapts to engine variations. The real-time optimization algorithm has three modes: maximum thrust, which maximizes excess thrust during accelerations, climbs, and dashes; minimum fuel, which minimizes fuel consumption during cruise; and the minimum FTIT (Fan Turbine Inlet Temperature), which extends engine life by reducing FTIT. The control function is adaptive and automatically adjusts to account for engine and aircraft variations, flight condition, engine deteriorization, and aircraft stores. This is accomplished by estimating the operating characteristics of the engine, inlet drag, and horizontal tail trim drag. Flight test showed significant benefits from this integration: an increase of thrust in the range of 9 to 15%, a reduction in fuel flow of 2% at constant thrust, and an estimation of a 50% reduction in high-pressure engine wear when temperatures are reduced by about 80°F.

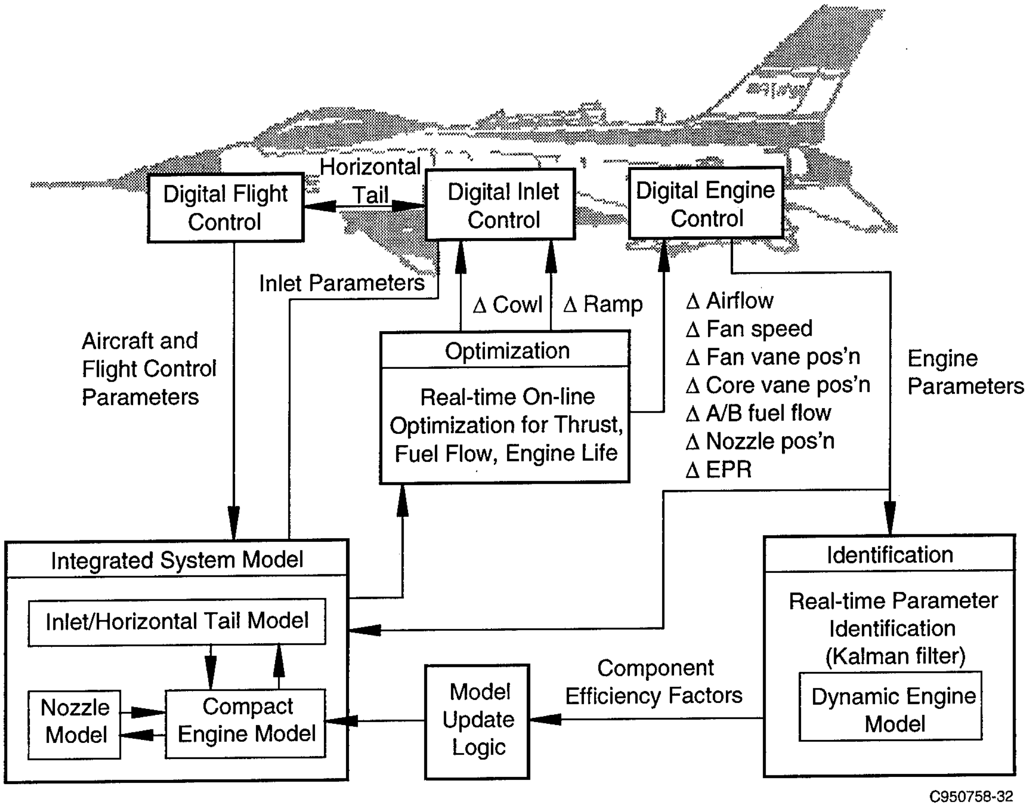


Figure 3-2. Performance-Seeking Control System

3.3.4 Pilot Information/Data Fusion

As aircraft systems have increased in complexity, so has the need to simplify pilot functions. The solution to alleviating pilot overload has been to automate as many of the noncritical and non-value-added functions as possible. Although automating many functions and actions has helped reduce pilot workload, still more had to be done. Information fusion and data fusion are forms of functional integration that continue to evolve to help resolve the pilot workload problem. In data fusion, which has been employed in navigation systems for many years, data is taken from many sources (usually various subsystems) and mathematically combined to improve a needed function. In the navigation system example, the data is taken from the IMU, altimeter, airspeed indicator, air data system, etc., and processed by a filter (Kalman, usually) to produce improved position information. Data fusion has been used to combine data from many sensors, such as radar and IRST, and then produce a single, highly accurate information packet to the pilot about the position, direction vector, and even the type of approaching aircraft (obtained from processing skin vibration features). Information fusion is also a form of integration, but one that usually results in complex data being presented to the pilot in a greatly simplified form. An example of information fusion is where complex numeric data weapons status is shown in pictorial format on the pilot displays. Both forms of fusion have been proven to be extremely effective methods for reducing pilot workload, increasing pilot situation awareness, and increasing pilot/aircraft system effectiveness.

3.3.5 Intership Data Integration

Whereas most concepts of functional integration are focused on the integration of functions within an aircraft, there is one significant program that is being sponsored by the USAF, the U.S. ARPA program office, and the French Ministry of Defense to define, develop, and demonstrate a distributed (multiaircraft) control system design that will support intership, cooperative attack, and defensive operations among a group of tactical fighters. This program, called the Control Automation & Task Allocation (CATA) program, is a concept in which operational decisions and even aircraft control are shared among a group of aircraft. This complement of aircraft can include fighter aircraft, lethal unmanned vehicles (LUVs), and other battle management aircraft (AWACS, JSTARS, ABCCC). Simulation of such a concept has conclusively demonstrated that aircraft attrition can be significantly reduced and mission effectiveness greatly improved in: (1) lethal suppression of enemy air defense operations, (2) defensive counter air operations, (3) time-critical high-value fixed targeting (airfields, bridges, etc.) operations, (4) critical mobile targeting (i.e., Scuds) operations, and (5) offensive counter air operations.

The functional integration technologies being used and developed for this intership concept design include automated and intelligent aircraft control, multiship flight management, autonomous operations, interactive databases, and cooperative planning. Because the aircraft share sensor data, situation assessment data, and data for controlling one aircraft from another, there is a large amount of information to process. These integration technologies require other critical technologies for the system to operate effectively, such as very high speed computing, secure and covert communications, and advanced video and data compression.

The CATA program is scheduled for completion in the year 2000. When deployed, it will be available for use in existing aircraft as well as future, new developing aircraft.

3.3.6 Integrated Control Law Evaluation (ICLE)

The Integrated Control Law Evaluation (ICLE) [3.2] program was one of the early efforts for examining integrated control modes and architectures for implementing these integrated modes into a VMS system. The approach was to define a fighter class vehicle with a set of requirements. Models for the vehicle systems were developed and the system was partitioned into a set of local and integrated modes. The four major integrated control modes were IFPC, crew protection, tactical flight management, and utility management. The subcategories of each of these modes were:

- IFPC
  Basic
  Increased Stall Margin
  PSC

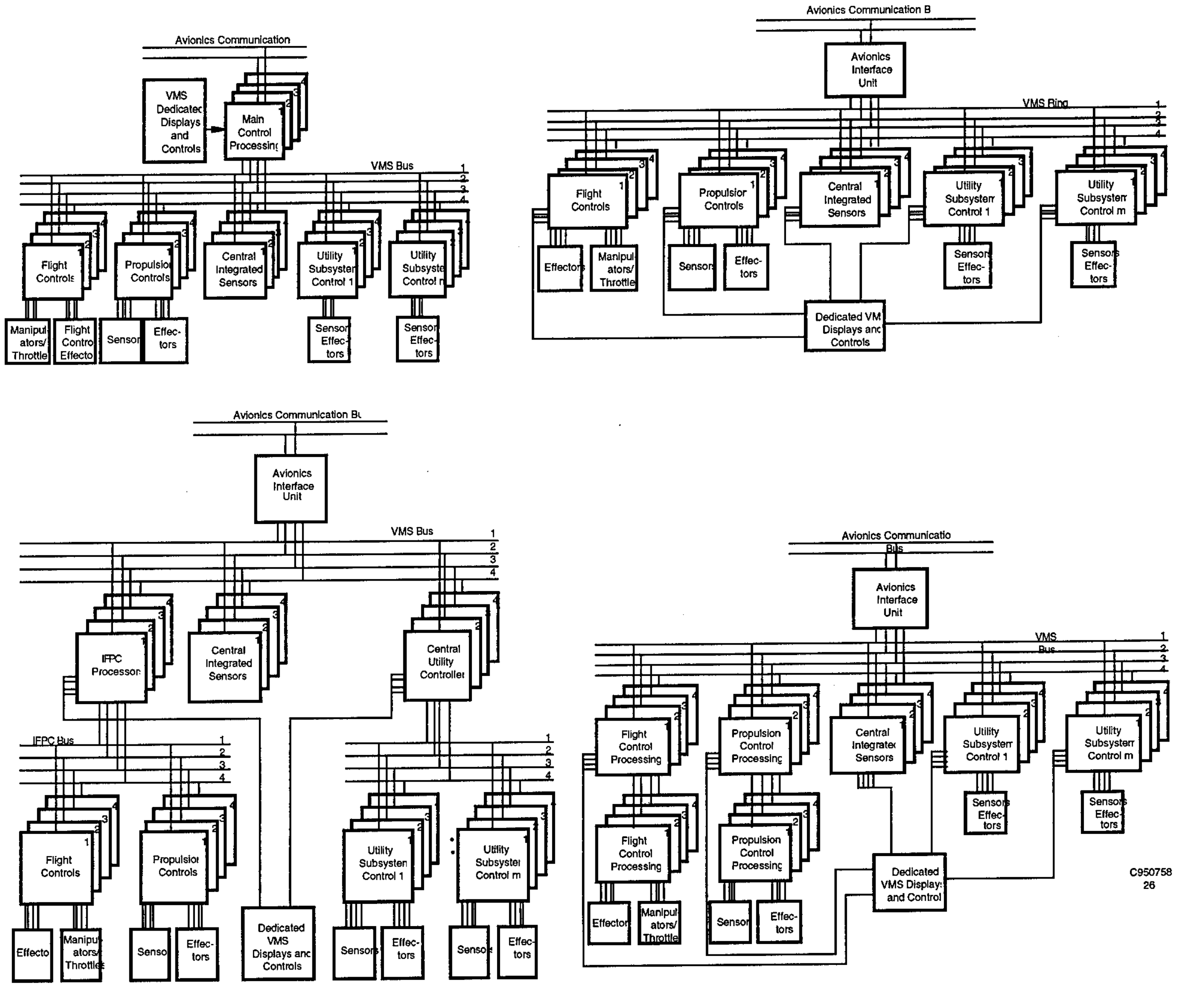
- Crew Protection
  Anticipatory Anti-g Suit Control
  Ground Collision Avoidance System
  Loss of Consciousness Detection

- Tactical Flight Management
  Integrated Fire Flight Control
  Terrain Following Terrain Avoidance
  Automatic Missile Evasion
  Return to Base Autoland
  Optimal Fuel/Time Trajectory

- Utility Management
  Closed-Loop ECS
  Active CG Control
  Variable-Pressure Hydraulic Control
  Smart Bingo Fuel Warning
  Electrical Load Management
  Fuel/Thermal Management
  Anticipatory Control for Power Transient Suppression

The processing requirements for this VMS were estimated as 4.5 MIPS of MIL-STD-1750 instructions and 400K words of memory. Architectural design issues were then established. These issues included: data communication method, control task partitioning, distribution of processing resources, interfaces to sensors and effectors, function synchronization, fault detection and isolation, reconfiguration, redundancy levels, pilot interface, packaging, maintainability, and supportability.

To examine the architectural issues, four VMS architectures were developed, Figure 3-3.



Avionics Communication
VMS Dedicated Displays and Controls
Main Control Processing
VMS Bus
1
2
3
4
Flight Controls
Propulsion Controls
Central Integrated Sensors
Utility Subsystem Control 1
Utility Subsystem Control m
Manipulators/ Throttle
Flight Control Effectors
Sensors
Effectors
Sensors Effectors
Avionics Communication B
Avionics Interface Unit
VMS Ring
Flight Controls
Propulsion Controls
Central Integrated Sensors
Utility Subsystem Control 1
Utility Subsystem Control m
Effectors
Manipulators/ Throttle
Sensors
Effectors
Dedicated VMS Displays and Controls
Avionics Communication Bus
Avionics Interface Unit
VMS Bus
IFPC Processor
Central Integrated Sensors
Central Utility Controller
IFPC Bus
Flight Controls
Propulsion Controls
Utility Subsystem Control 1
Utility Subsystem Control m
Effectors
Manipulators/ Throttle
Sensors
Effectors
Dedicated VMS Displays and Controls
Avionics Communication Bus
Avionics Interface Unit
VMS Bus
Flight Control Processing
Propulsion Control Processing
Central Integrated Sensors
Utility Subsystem Control 1
Utility Subsystem Control m
Sensors Effectors
Dedicated VMS Displays and Control
Effectors
Manipulators/ Throttle
Sensors
Effectors
C950758
26

Figure 3-3. ICLE VMS Configurations

The major discriminators among them are task partitioning options, processing distribution, data communications, and packaging options. The four architectures varied from centralized to distributed and contained the complement of integrated control modes. A ground-based simulation, including a cockpit for man-in-the-loop evaluations, was established. Results showed that a centralized approach eased the integration of control modes and reduced the bus traffic because data exchange is reduced. However, V&V of the system becomes more complex because the controls modes are highly coupled. The distributed system has higher bus traffic, but the V&V process is eased because of the separation of functions. The other configurations studied were combinations of the distributed and centralized approaches.

3.3.7 Advanced Vehicle Management System (AVMS)

The AVMS program defined the candidate VMS architectural concepts for advanced vehicles and defined an integrated tool environment necessary for development of an advanced VMS system. The program first determined the functional requirements for a multimission fighter and then a baseline VMS architecture was developed. Six additional VMS architectures were then defined and compared to the baseline. The criteria used to perform this comparison consisted of 14 major categories (redundancy, number of LRUs, number of LRMs, flight safety, LCC, survivability, weight, availability, performance, risk, V&V, installation, power, and growth potential). Several of these categories had subcategories. LCC included development cost, acquisition cost, and O&S cost. Availability included reliability, maintainability, and supportability. Performance was broken down into processing speed, communications, efficiency, and data latency. Last, installation was split into volume, cooling, and environment.

The VMS functions considered in the evaluation are shown in Table 3-3 below.

Table 3-3. AVMS Functions

Flight Control	**Propulsion Control**	**Utilities Control**
Local Trajectory Generation Airframe Control Pilot/Vehicle Interface Kinematics/Inertial Reference	Engine Control Nozzle Control Air Inlet Control	Environmental Control Fuel Control Hydraulic Control Electrical Control Mechanical Control Auxiliary Functions
Local Fault Reporting and Recording		

The architecture that was rated the highest is shown in Figure 3-4. The architecture is triplex and uses three vehicle management processors (VMPs) and three vehicle reference units (VRSs). An optical data bus is used to couple the VRSs, VMPs, and vehicle interface units (VIUs).

A VMS development methodology was also defined. This process had the following characteristics: top down, requirements driven; disciplined and repeatable; incorporates concurrent engineering principles; allows for design iteration and refinement; and incorporates balanced tradeoff analyses. The requirements for an integrated tool environment based on this process were then developed.

3.3.8 Subsystem Integration Technologies (SuIT)

The SuIT program [3.5] was initiated to assess integration opportunities for the aircraft utility systems. Systems to be integrated include fuel management, environmental control, hydraulics, gas generation, and auxiliary power generation. These subsystems have been developed independently and optimized at the subsystem level. The SuIT

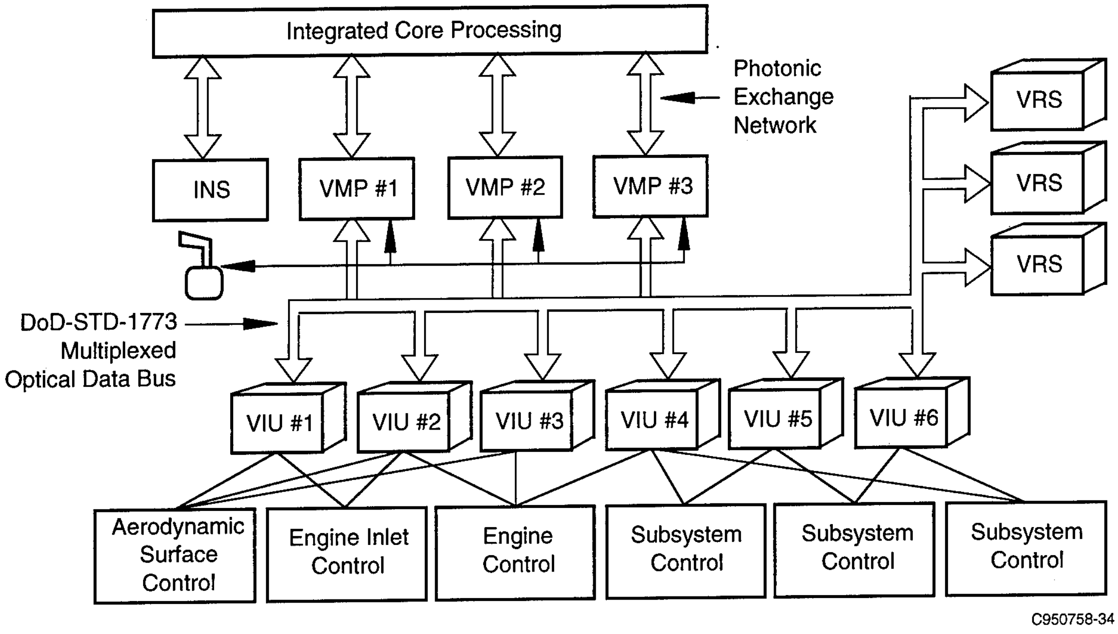


Figure 3-4. AVMS Architecture

approach is to design an integrated system with optimization occurring at the system level. When integrated with the aero/propulsion systems, it is expected that benefits can be achieved in the areas of energy management, aircraft performance, load shedding, power on demand, cooling on demand, diagnostics, LCC, and increased fault tolerance. The approach is to maximize the use of common or shared hardware and fluids to perform required functions, use waste energy, and provide better overall system energy management. Figure 3-5 shows a SuIT configuration compared with the traditional approach.

3.3.9 Integrated Closed Environmental Control System (ICECS)

The ICECS program [3.4] ground demonstrated a closed environmental control system to demonstrate reduced LCC and performance benefits compared to current technology. The ICECS system consisted of: a closed vapor cycle refrigeration loop, heat rejection transport loops, two closed and one open avionics heat transport loops, a cabin heat transport loop, and a digital control system. The F-15A was used as a baseline aircraft for making comparisons. Goals were to design a system that would deliver cool, clean, dry, and constant-temperature heat sink media to the various heat loads, minimize bleed air usage, minimize power extraction, and minimize weight. A detailed nonlinear simulation was first established to predict both steady-state and transient performances based on actual mission profiles. The simulation was then followed by a ground based demonstration. The area of most interest to this report is the control system, as it would be that portion of the ICECS system contained within the IVMS. Figure 3-6 shows the functional block diagram of the control system. The system utilizes a multi-input, multioutput dynamically decoupled structure to minimize control interactions among the control channels. The supervisory control has as inputs: altitude, mach number, ram air temperature, climb rate, engine throttle position, and heat loading information. The outputs are gains, effector positions, and set points. The supervisory control also performs BITE and reconfiguration functions. Results of the program demonstrated that a 16% reduction in life-cycle cost is achievable due to an improved avionics environment. This number was based on analysis. A 53% reduction in gross take off fuel penalty is also achievable due to reduced bleed and ram air consumption. If the aircraft TOGW remained constant, this quantity of fuel would extend the aircraft range by 35 mi. Because of the improved reliability of avionics afforded by lower, more stable cooling, 16% reduction in LCC is projected.

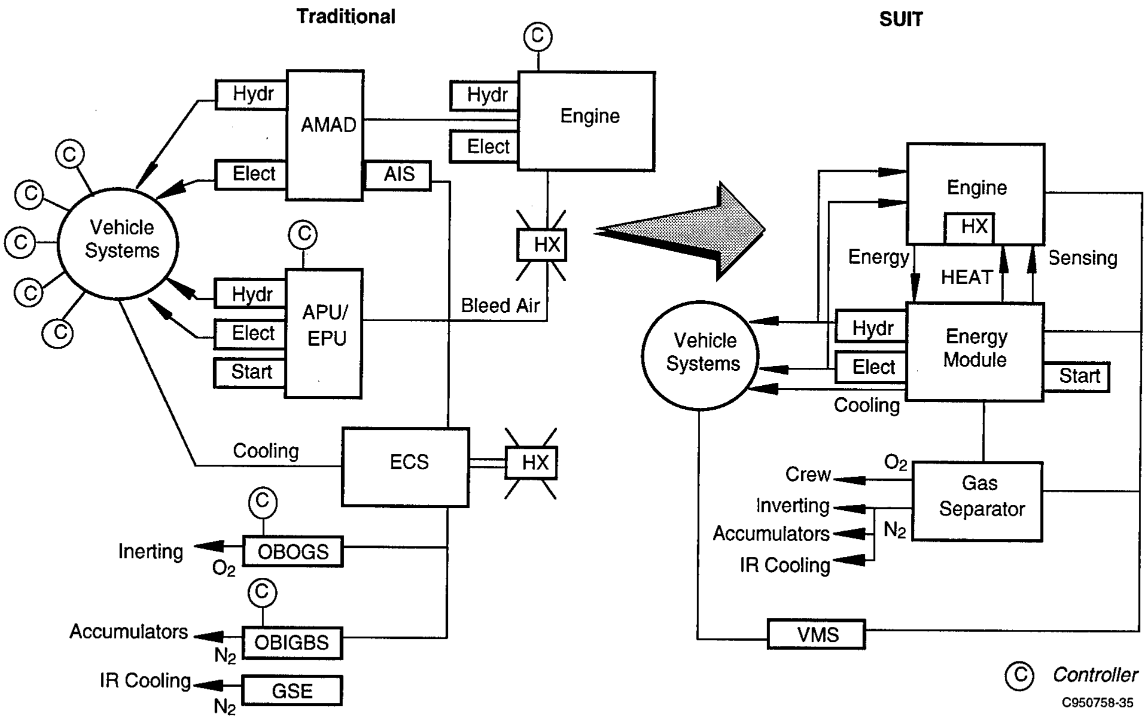


Figure 3-5. SuIT Component Reduction

3.3.10 Fault-Tolerant Designs

System survivability is a prime concern for both military and commercial aircraft. Fault tolerance is thus required by systems to assure that the system will operate through the mission and flight safety will not be jeopardized. The highly integrated nature of the IVMS provides an excellent framework for increased fault tolerance. The physical integration of the aircraft systems through the data bus network makes information and computational sharing available to all subsystems. The dispersed, multipath distribution system can be designed to allow the loss of several sensors and computational resources and still operate without loss of flight safety. The concept design should be able to be extended to provide a system that need not require unscheduled maintenance but would require maintenance on a scheduled basis only. Such a system would decrease support costs.

The main features of the fault-tolerant system are the ability to detect a fault, isolate it, and reconfigure the system. The reconfiguration could be through the replacement of physical resources or through algorithmic means.

3.3.11 Pooled Spares Designs

The "pooled spares" aircraft systems concept was fully exploited in the USAF PAVE PILLAR program. This integrated digital avionics system utilized the emerging processing power of VLSI technology. The system provided the hardware necessary to implement many functional features by using a small family of modular, line-replaceable units to accomplish almost all of the signal and data processing functions. Historically, many of these functional features had been cost-prohibitive. These features included multiple sorties without maintenance, extremely high mission reliability, and the reduction of field and maintenance support.

The PAVE PILLAR program proved that it was advantageous to convert as many of the avionic sensor functions from analog to digital design as possible. By doing so, it was possible to increase the level of reliability and fault tolerance even more than if only the central processing systems were arranged in a pooled spare

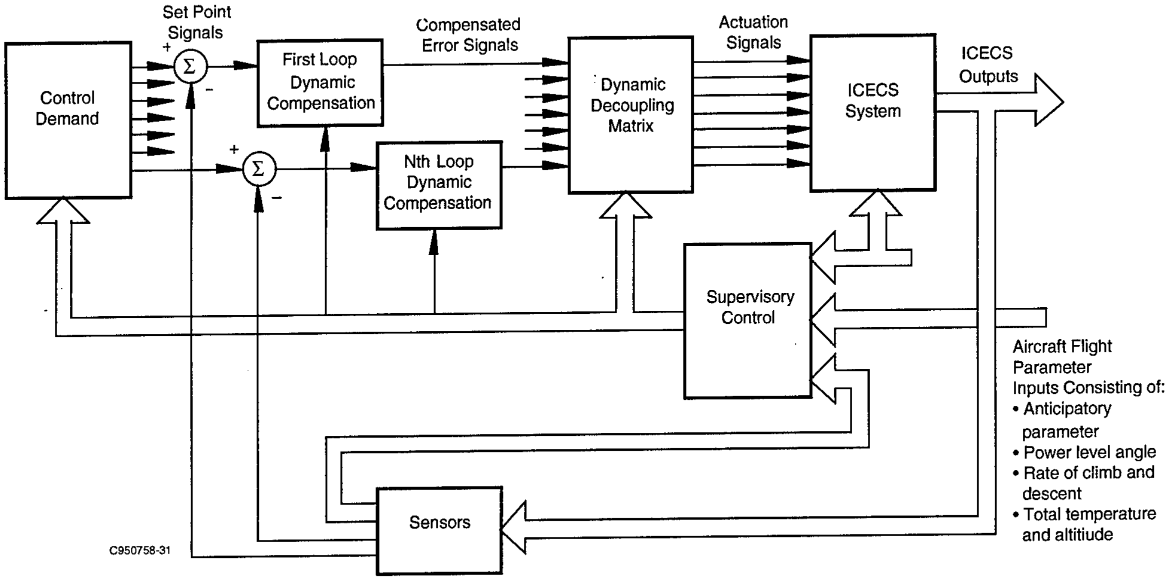


Figure 3-6. ICECS Control System

configuration. It was also possible to reduce the proliferation of signal and data processors, which tended to reduce the initial cost of the system. Performance was also enhanced by the tight coupling of both signal and data processing assets, because the combination allowed new capabilities in data fusion to improve situation awareness. Weight and volume were also substantially reduced due to packaging advances.

The practicality of applying the PAVE PILLAR concept to the IVMS has also been investigated. It has been determined that the basic modular processing elements are useful in IVMS design as well. However, new innovative actuators, sensors, and controllers that have common digital elements will have to be developed before the pooled spares technology can be fully exploited in IVMS designs.

3.3.12 Common Module Designs

Common modules with carefully selected functions and features are an essential element in the design of reliable, low-cost avionic and vehicle management systems. They provide the building blocks for the "pooled spare" concepts (see Subsection 3.3.11, above for discussion), fault-tolerant designs (Subsection 3.3.10), and automated maintenance/diagnostic designs (Subsection 3.3.13). Thirteen modules were identified for the pooled spare concept that would provide all the functionality needed to support all the electronic functions in most aircraft. Although few of the modules have been built and used, the cost-effectiveness has been proven. The common module, when designed to be self-monitoring for faults, can be connected in ways that allow the system to be automated for fault detection and system diagnosis. Equally important, the common modules can be also connected in ways that permit the redistribution of functions after a failure occurs, which leads to fault-tolerant systems.

Many programs have attempted to exploit the common module concept. Most of these programs resulted in substantial cost savings and have proved to be an effective means of increasing the effectiveness of the aircraft system. The MIL-STD-1553 data bus and the MIL-STD-1750 processor are examples of modules that have proven to be cost-effective. The advantages of using a set of common modules in aircraft are many. For example, modularity greatly simplifies maintenance on the flight line and on forward bases. With systems constructed in definitive modules, maintenance is greatly simplified (remove and replace). Furthermore, the experience and knowledge level of the field maintenance personnel need not be as high as needed to repair a complex, highly integrated (physically) system. Also, the cost of spares can be lowered, and the number of spares at a location can be lowered as well. However, improper selection of what functions and what level (amount of functions or equipment complexity) should be included in the modules can cause unacceptable consequences. This important lesson was learned during the early years (late 1950s and early 1960s) of the U.S. military Standard Hardware Program (SHP). The SHP divided most of the aircraft electrical and electronic subsystems into common modules, but the modules were developed with too little functionality (i.e., an operational amplifier). As a result, the number of modules proliferated until the advantages of using common modules in aircraft was lost. The most serious shortcoming, however, was the inability of the program to adapt to the rapid changes that were occurring in electronics during those years. Although technology advances could allow more functions to be incorporated on a single chip (or a module), and the additional functions could be obtained at a significantly lower cost, the existing common modules were in the inventory and could not be easily replaced with better, lower cost modules. Soon after these problems were understood, changes were made that eliminated many of the problems; modules with little functionality were eliminated, and modules with much more functionality were selected. Due to these and other lessons, most common module programs, such as the standard MIL-STD-1553 data bus and the MIL-STD-1750 processor, have avoided the problems associated with common modules. This has been accomplished by carefully selecting the scope of the module functions, clearly identifying and controlling the module interfaces, and maintaining the interchangeability (form, fit, function) features of the common modules.

3.3.13 Automated Maintenance/Diagnostics

One of the more significant payoffs of the IVMS may be in the area of automated maintenance/diagnostics. As the system is highly integrated, a wealth of information is available to aid in the detection, isolation, and identification of failures and faults. This function can be performed by built-in test within a particular subsystem and inference techniques which air in the process. The goal is to provide the capability to auto fault isolate to a line-replaceable unit. The storing of faulted information during flight is facilitated by the centralized nature of the system. This should lead to a system that requires no or a minimum of additional equipment by maintenance personnel.

3.3.14 Redundancy Management

Redundancy management requirements are driven by survivability and availability requirements. The level of redundancy required for a particular subsystem contained within the IVMS is a function of the criticality of the function being performed by that subsystem. For example, flight and propulsion control must function full time to assure vehicle safety. However, an environmental control system fails, certain mission critical systems can be shut down and not affect the vehicle safety. Although, the loss would undoubtedly result in a mission abort. As an extreme, the redundancy design of the total IVMS could be mechanized at the level of the most critical function; however, it would result in a very costly system. The appropriate design usually evolves to a mixed redundancy approach that takes into consideration the criticality of each one of the IVMS functions.

3.4 ASSESSMENT OF IVMS DESIGNS

3.4.1 Assessment of Current VMS Designs

Over the past ten to fifteen years, various integrated vehicle control functions such as IFPC have been demonstrated and have shown quantifiable performance benefits resulting from functional integration. Recently, vehicle utility systems have been integrated, and the resulting systems have shown excellent payoffs. These systems include the SuIT and the ICECS. Various studies, such as the ICLE and AVMS, integrated both the vehicle control functions and the utility systems. These studies have postulated benefits from the development of IVMS systems; however, benefits at a vehicle level, for the integration of the total set of technologies, have not been quantified to the same level. In part, the quantification of the total IVMS has been the lack of a suitable baseline to provide the basis for comparison of benefits. The usual process has been to take an existing aircraft, and then apply current or projected technologies to current systems. A VMS is then defined with the same technologies so that benefits are not always apparent.

3.4.2 Metrics for Assessing IVMS Designs

The integrated systems referenced above are only partial IVMS systems since each one does not include all of the IVMS functions. The question then arises as to what additional benefits can be achieved by the IVMS approach. A need then exists for assessing total benefits and/or penalties beyond what has been currently achieved.

Table 3-2 shows many of the metrics that can be used to assess the relative benefits and penalties of an IVMS. Depending upon which enabling designs are being considered, the metrics shown should provide guidance as to the advisability of the design, and a measure of how well the design satisfies the vehicle and the IVMS design requirements.

Any IVMS benefits/penalties analysis must be approached from a total vehicle level to provide a fair assessment of its capabilities. The analysis must also consider a specific case using an appropriate baseline system to add credibility to the results, since an IVMS will invariably change from application to application.

3.5 ASSESSMENT METHODS

The IVMS technologist has three methods at his disposal to evaluate the benefits and/or penalties associated with an IVMS system design. The methods include analysis, simulation, and flight test. The method utilized depends on the portion of the design cycle the effort is in and what parameter(s) are being evaluated. These methods can be used to assess the requirements of performance, survivability, affordability, and logistics as indicated in Table 3-4.

Table 3-4. Benefits Assessment Methods

	Analysis	**Simulation**	**Flight Test**
Performance		X	X
Survivability	X		
Affordability	X		
Logistics Test	X	X	

3.5.1 Analysis

Analysis is usually the means used to assess survivability, affordability, and logistics. Since survivability consists of two parts, (1) vulnerability - the aircraft's ability to withstand damage, and (2) susceptibility - the measure of the amount of that the aircraft is subject to attack, an analysis of survivability depends on models that generate aircraft geometry descriptions and shot line intersections. From these models it is possible to determine the amount of penetration of foreign objects into the aircraft, and the resulting probability of kill.

Analysis of affordability considers estimates of all of the elements of Life Cycle Cost (LCC), which includes both development and O&M costs. The analysis looks at: the cost of components or LRUs per aircraft, the weight (cost to carry the LRUs in terms of fuel) of the LRUs, the cost of developing (or reusing) the LRU, the field mean time to repair, and the cost of maintaining spares in the inventory.

Analysis of logistics considers estimates of: maintainability (MTTR), maintenance personnel required, and operational availability (MTBMA, and number of sorties without maintenance).

3.5.2 Simulation

Simulation is used to expose the IVMS to as complete a set of possible environments as might be encountered in operational service. This includes initializing the IVMS in a wide variety of degraded states, and inclusion of a range of fidelity in actuator and sensor models. Simulation is very valuable for assessing and validating the performance, providing a comparison between analytical predictions and simulated performance. It can not be over-emphasized that the simulation results are dependent upon the accuracy of the environmental models, including both the vehicle's environment and the airframe and propulsion components as well. It also is dependent upon the fidelity cues presentation to the pilot, so that the pilot's inputs to the IVMS are equivalent to that which would occur under the similar circumstances in an operational setting.

A generalized view of the major environmental components that interface with the IVMS is presented in Figure 3-7. These components are modeled to varying degrees of fidelity to simulate the IVMS submerged in an environment representative of that anticipated in operational flight. Table 3-5 provides a characterization of two levels of fidelity, with comments on each of the environmental components. The components are subdivided into those internal to the airframe, external to the airframe, and the pilot or crew. Although not shown in the table, there are additional simulation components that may be involved, although not interfacing directly with the IVMS. For example, in very high fidelity simulations, a moving base cab may be included to provide the pilot with representative motion cues.

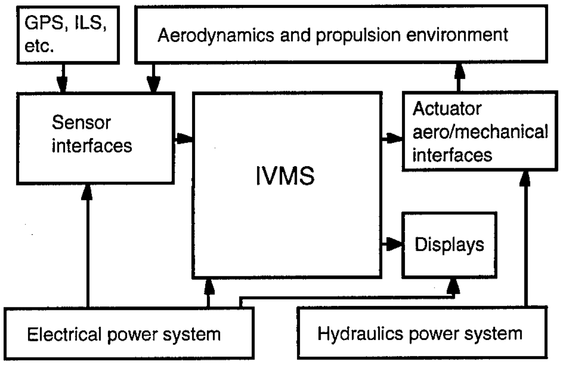


Figure 3-7. IVMS Functional Interfaces

In many cases, simulation is the best and most cost-effective means of obtaining test data. In cases where statistically valid results must e obtained, Monte-Carlo simulations can be run which change combinations of variables in a structured, but random way. Simulations are valuable in achieving some the extreme combinations of environmental data, cases in which it would be difficult or impossible to set up the same combination of variables in a flight test setting.

3.5.3 Flight Test

Flight test exposes the IVMS in its "verified-for-flight" state to important real-world conditions. This is accomplished by replacing the high fidelity environment models with the actual environment. The pilot actions are the result of the responses to real world cues. In those cases when the simulation results are validated by the flight test results, then confidence in the entire set of simulation results is warranted. In those other cases when there are disagreements between the simulation and flight results, then the reasons for the disagreements must be determined. Frequently, the reason is traceable to inaccurate environment models in the simulation. Furthermore, in a complex IVMS, the deficiencies in the environment model do not become apparent until some of the interactions are brought out by the IVMS.

In a similar way as in the previous section, a characterization of the various environmental components is given for flight, as compared with a high-fidelity ground-based simulation. This is done in Table 3-6.

Beginning with the components internal to the airframe, the various sensors are exposed to the actual combination of conditions consistent with the real-world flight environment. Pneumatic air-data sensors, for example, may encounter local flow effects in the test aircraft in flight that were not modeled correctly in the simulator. The actuators, in flight, may be involved in an aerodynamics-induced buzz condition that cannot easily be simulated in a ground-based setup. The test airplane in flight provides an accurate excitation of the airframe-mounted sensors, including flexible-body modes, sensor-mounted sensors, including flexible-body modes, sensor-

Table 3-5. Simulator Environments Relative to the IVMS

Internal to Airframe	**Sensors**	**Actuators**	**Airframe**	**Subsystems***	**Pilot/Crew**
Simplified simulation	Digital or analog simulation	Digital or analog linear models	Assumed rigid body (no effect)	Static conditions or wall power	"Canned pilot" or engineering pilot
High-fidelity simulation	Actual sensors	Nonlinear models	Representative flexible body modes	"Iron-Bird" or brass-board simulation	Test pilot or full crew

External to Airframe	**Aerodynamics/ Propulsion**	**GPS, ILS, etc.**	**Air/Ground IF, Taxi**
Simplified simulation	Uncoupled models (<6 DOF equations)	Ideal signals	Simple computer models
High-fidelity simulation	Full non-linear (6 DOF equations)	Simulated radiated signals	Complex computer models

* Note: Subsystems include hydraulics, electrical power, etc.

Table 3-6. Flight Environment Relative to the IVMS

Within the Airframe	**Sensors**	**Actuators**	**Airframe**	**Subsystems***	**Pilot/Crew**
High-fidelity ground-based simulation	Actual sensors	Non-linear models	Representativ e flexible body modes	"Iron-Bird" or brass-board simulation	Test pilot or full crew
Actual conditions (Flight)	Actual sensors in real world	Actual hardware	Test airplane in flight	Test airplane in flight	Test pilot or test crew

External to Airframe	**Aerodynamics/ Propulsion**	**GPS, ILS, etc.**	**Air/Ground IF, Taxi**
High-fidelity ground-based simulation	Full non-linear (6 DOF equations)	Simulated radiated signals Ideal signals	Complex computer models
Actual conditions (Flight)	Real world when conditions met	Real world signals	Real world conditions

*Note: Subsystems include hydraulics, electrical power, etc.

mounting compliances, and other factors that may interact with the IVMS in a significant way. The hydraulic and electrical power sub-systems may have some differences between ground simulation and flight, although a good "iron-bird" simulator generally provides an accurate reproduction of the flight vehicle.

The test pilot or crew can be assumed to be the same or equivalent, so the differences, if any between flight test and simulators can be the results of differences in cues or physiological factors. The most notable differences may manifest themselves in pilot induced oscillations that occasionally occur in flight, whereas they are not encountered in ground-based simulators. One example is when the flexible body modes couple with the pilots' arms or torso in flight, as well as compensatory feedback commands, which in turn couple with the flight control system and vehicle rates and attitude responses. In this case, the IVMS would be responding correctly to the set of excitations, but the total vehicle-system-pilot in-flight response is an unacceptable oscillation.

Continuing on to the "external-to-the-airframe" components, the aerodynamics or propulsion characteristics may be different from the simulation models. This can be in a new flight regime, such as air-breathing hypersonic propulsion, where very little test data exists representative of the actual full-scale Reynolds numbers environment.

In another "external-to-the-airframe" component, off-airframe signals need to be provided as excitation to the on-board pickup sensors. There may be factors of importance, such as orientation of the aircraft, which may not have been adequately modeled in the ground-based simulation.

The final "external-to-the-airframe" components of interest are some of the non-flight inputs, such as ground roll-out and high speed taxi. The air-ground transition has frequently been a source of flight test surprise. Many times, weight-on-wheels switches are exposed to different load paths in the actual flight situation as compared with high-fidelity ground-based simulations. This can result in unusual mode states falling outside of the design space, yet experienced in flight test. Another problem area brought out in flight test is the excitation of flexible body modes due to runway roughness, and coupling with the IVMS. Frequently, the magnitudes and effects of these couplings are far more pronounced in flight test than in any of the corresponding ground-based simulations.

3.6 CONCLUSIONS AND RECOMMENDATIONS

The benefits of various VMS control modes and architectures have been demonstrated over the past several years through the use of analysis, simulations, laboratory demonstrations, and flight test. Most of these efforts have shown improvements in performance, system capability, reliability, and LCC. New control modes and architectures could not be incorporated into actual aircraft design without an analysis and demonstration. This was necessary to prove that the benefits clearly outweighed the penalties. It follows that analysis and proof of concept demonstrations for IVMS is required as well. However, tools and analysis methods for testing highly integrated vehicle management systems will need to be developed, since traditional functions may not be separately analyzed (or tested) within the system context. For this reason, it is also likely that it will require more V&V testing to assure the desired safety of flight in an IVMS than that required for existing VMS designs.

3.7 REFERENCES

[3.1] Stewart, J. F., Burcham, F. W., Gatlin, D. H., "Flight-Determined Benefits of Integrated Flight-Propulsion Control Systems," NASA Technical Memorandum 4393, June 1992.

[3.2] Chen, W. Z., Stein, L. J., Haiges, K. R., "Vehicle Management System Architectural Considerations," 8th Digital Avionics Conference, San Jose, California, 17-20 October 1988.

[3.3] Bedoya, C. A., Mohr, J. L., "An Advanced Vehicle Management System," SAE Aerospace Conference and Exposition, Dayton, Ohio, 20-23 April 1995.

[3.4] Edgar, J. M., Campbell, B., "Evaluation and Control of an Integrated Closed Environmental System (ICECS)," 20th Intersociety Conference on Environmental Systems, Willamsburg, Virginia, 9-12 July 1990.

[3.5] Burkard, A. H., Haskin, W. L., "Concepts for Aircraft Subsystem Integration," SAE Aerospace Atlantic Conference and Exposition, Dayton, Ohio, 20-23 April 1993.

CHAPTER 4

ARCHITECTURE AND IMPLEMENTATION

4.1 TYPICAL FUNCTIONS TO BE INTEGRATED

As discussed in Chapter 1, vehicle management systems implement a variety of functions. Initial attention has concentrated on the integration of such functions as flight and engine control. Other functions can be integrated as the techniques and components to support integration mature. For manned aircraft, the functions associated with vehicle management as distinct from mission management include those listed in Table 4-1.

Table 4-1. VMS Functions

Flight Control
Engine Control
Air Data
Fuel Management
Secondary Power Control Electrical Generation Battery Monitoring Hydraulic Generation
Utilities Control Undercarriage Environmental Control System Brakes Steering Life Support Systems Fire Detection and Extinguishing

4.2 IVMS ARCHITECTURE REQUIREMENTS

Vehicle systems are integrated for a variety of reasons, as discussed in Chapter 3. To achieve the advantages of integration, the VMS architecture should satisfy the following requirements:

- Share data and control between functions to gain operational advantage;
- Sufficiently partitioned to guarantee that errors/faults do not propagate;
- Enable different functions to be implemented at different redundancy levels and thus match the architecture with the reliability and availability targets;
- Ensure that functions interact so as to provide the features the system requires;
- Minimize coupling between functions to simplify development;
- Reduce the delays in data handling;
- Facilitate a crew interface that is intuitive and minimizes the loss of functionality when failures occur;
- Allow common subfunctions to be shared to reduce the costs of implementation.

Clearly, many of these requirements conflict, and it is the designer's task to develop an acceptable architecture.

4.3 FACTORS THAT INFLUENCE THE ARCHITECTURE

Vehicle management systems can be integrated in a variety of ways. The factors influencing the architecture include:

- The size and complexity of the functions to be implemented and integrated;
- The degree of fault tolerance, safety, reliability, and survivability required from the system;
- The number of processors required to implement the functions;
- The form of interfaces with communicating systems;
- The maintainability requirement.

As described in Chapter 2, two main forms of integration can be considered:

- First, the functional level that defines how the functions are partitioned and how they interrelate. The functional level will also determine how the functions are to be decomposed into processes and how the processes are to be linked and controlled.
- Second, the physical level that defines how the functions are implemented by hardware and software components. The physical level will determine the arrangement of hardware and software components required and the form of intercommunication between those components.

Both the functional and physical architectures are greatly influenced by the safety, reliability, survivability, and availability requirements. The requirement to tolerate the effect of random hardware failures or software error has a major impact. If the system is required to be fail-safe or fail-operational after a random hardware failure, multiple redundant lanes of hardware are required. This restricts the types of functions that can be realized and leads to the transfer of data between otherwise separate areas.

If the system is required to survive software errors, other techniques are required; for example, multiversion software, dissimilar hardware, or monitoring functions.

4.4 FUNCTIONAL ARCHITECTURES

The functional architecture of a system can be described by the form of partitioning used to achieve the integrated functions and the type of control.

Forms of partitioning include:

- Monolithic—no partitioning,
- Data-flow-oriented partitioning,
- Functional partitioning,
- Redundancy level oriented,
- Iteration level oriented,
- Object oriented.

Forms of control include:

- Hierarchical ,
- Federated,
- Autonomous,
- Cooperative.

4.4.1 Monolithic/No partitioning

The most basic form of functional architecture is monolithic. In such an architecture, there is no partitioning of functions and the data is global. Some early systems had monolithic structures, but they were hard to understand, difficult to monitor, and prone to errors. The lack of a clear structure led to errors in control flow and data handling. It was difficult to manage the development of systems of any significant size because of the lack of clear interfaces.

Fortunately, few current systems are built with a monolithic structure. Also, the additional complexity created by a monolithic structure makes it an unsuitable architecture for integrated vehicle management systems.

4.4.2 Data Processing Oriented

Many current systems are designed using data flow methods. The structured analysis methodologies, such as Yourdon, which are based on data flow diagrams, have led to a generation of systems that have architectures with sets of multilevel processes linked by data flows. The established methodology and the extensive and mature tool support make this approach to system development very attractive.

Although the processes are often chosen to represent system functions, there is an inevitable pressure to rationalize the data flows and the processes to create an architecture that is based on data flows rather than system functions. Indeed, some structured analyses create control structures based on data flows rather than the more intuitive system behavior.

4.4.3 Functional Partitioning

When functional decomposition methods are applied, an architecture is developed that maps closely to the required system behavior [4.1].

Such architectures are easier to understand and retain many of the partitions that ensure that functions do not interact unnecessarily. The architecture and components are easier to modify and verify; however, they may prove to be inefficient to implement because of the separation between functions that may hide connections/relationships that could be used to improve performance or avoid duplication of processing. It is difficult to optimize data flows and the associated throughput when functional partitioning is taken to low levels.

The performance disadvantages associated with functional partitioning have to be balanced against the major gain it affords in the management of complexity.

4.4.4 Partitioning Based on Redundancy

In many systems, the availability required from one function is significantly different from that required from another function. Often the pressures of cost or resources will dictate that each of the system elements match the associated requirement rather than overengineer the element with the lower target. When such systems are developed, it is necessary to partition them so that when the function with the lower level of resource fails, the system will reconfigure so that the more critical system continues to operate.

To ensure that this objective is met, the integrated system must be partitioned so that the processes dependent on the failed resources can be deactivated and the system continues operating with revised functionality. Typical examples include the use of estimated sensed signals after failure with remaining sensors.

4.4.5 Hierarchical Control

With hierarchical control, there are one or more levels of control and the higher level functions control a group of lower level functions, as shown in Figure 4-1. Examples include mission management aids in which the mission management functions control the lower level functions such as sensor configuration and flight control.

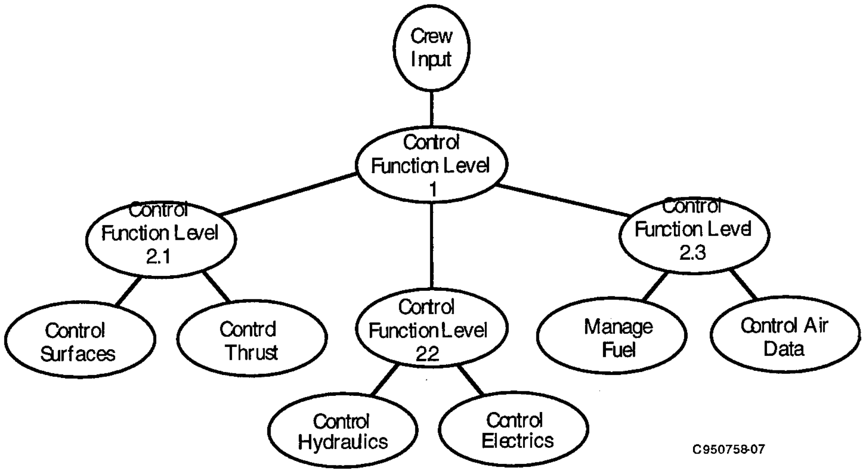


Figure 4-1. Hierarchical Control Structure

Such an architecture facilitates the overall control of the vehicle and allows decisions to be made with full information using the full capability of the vehicle. It improves the visibility of system operation because the high-level decisions are converted into lower level actions in a top-down structure. It also leads to coherence, as the functions respond to a single control.

4.4.6 Federated Control

As indicated in Figure 4-2, with federated control, the degree of control exercised at the top level is reduced and many control decisions remain at a local level. The federated functions respond to commands from the center but have the authority to meet the requirement in a locally determined manner. Examples include the way in which an engine responds to throttle commands from a flight management function. This architecture is one approach that can give the advantages of modular implementation without the complexity associated with total integration. Federated architectures reduce the complexity of the central function but with the disadvantage that the center may make unrealizable demands or full capability may not be achieved.

4.4.7 Autonomous Control

With autonomous control, the functions are segregated and the crew ensures that the vehicle operation is coherent, as shown in Figure 4-3. Many older aircraft have such an architecture. They have the advantage of simplicity and reduction of fault propagation.

The disadvantages include the demands of the crew who have to coordinate the functions and the loss of potential performance that a functionally integrated system can give.

4.4.8 Cooperating Control

Figure 4-4 shows that with cooperating control, the functions are linked together to achieve overall functionality. The individual functions respond to commands/information from the central control system but without knowledge of the total situation. The advantage is the partitioning of the functions, which eases the development of the individual functions and reduces the opportunities for fault/error propagation.

The disadvantage is the potential loss of functionality and the multiple number of system states that have to be investigated.

The object-oriented design method has many properties of cooperating functions.

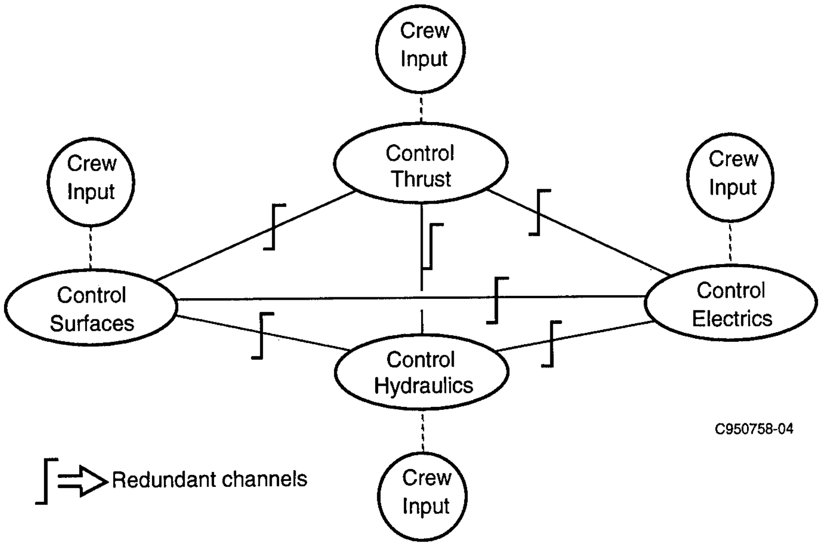


Figure 4-2. Federated Control Structure

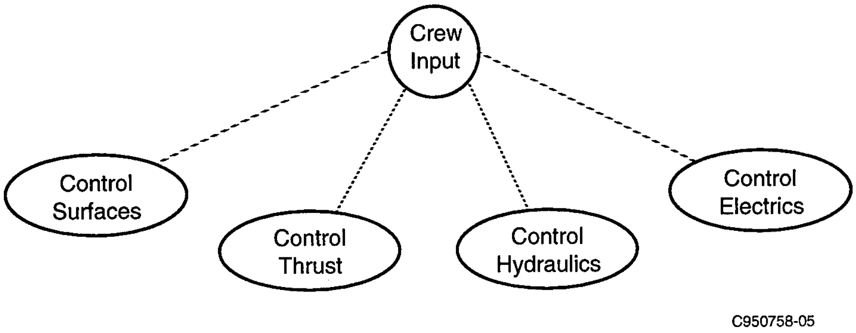


Figure 4-3. Autonomous Control Structure

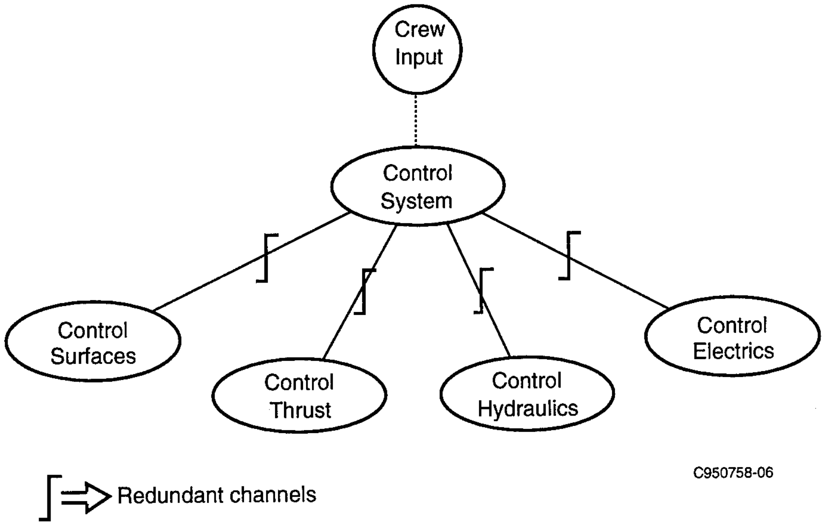


Figure 4-4. Cooperating Control Structure

4.4.9 Limitations of Integrated Functional Architectures

The major limitations of functional integration relate to the management of complexity. Strategies [4.2] and guidelines for the management of complexity have been developed; the major strategy is "divide and rule." The extent to which functions should be divided (or remain separate) depends on the design methods and tools that are available to support the development, verification, and certification of systems.

Many tools are now available to engineer the large number of requirements that integrated systems generate. Capture tools are now required to handle hundreds of thousands of requirements, and the latest versions of tools reflect this requirement by having virtually unlimited capability.

The modeling of behavior is even more important to ensure that the system will behave in a correct and predictable manner under all scenarios. Tools are becoming available to enable the designer to model and hence understand the behavior of complex systems; examples include Ascent Logic's RDD-100 and i-Logic's StateMate.

Even with these advances, difficulties remain in handling the increased functional complexity. The interaction generated by additional links between functions and the increasing sophistication of functions, as software-based systems go into the third or fourth generation of development, make it difficult to ensure completeness.

4.5 PHYSICAL IMPLEMENTATIONS

4.5.1 General

The physical architecture of integrated vehicle management systems is determined by:

- The resources required to implement the functions,
- The degree of redundancy,
- The physical separation between functions/lanes,
- The use of standard physical modules.

4.5.2 Architectural Partitioning

Clearly, the traditional architectures for integrated systems will require significant modification if safety-critical functions are to be integrated with noncritical functions. As a consequence, there will be an increase in the resources required to develop them.

Forms of partitioning include:

- Separate the top-level functions and limit the data flows between them;
- Allocate separate processes/processors for different functions;
- Allocate separate processors for each level of safety criticality;
- Allocate separate buses for each level of safety criticality;
- Partition the input and/or output processing from the noncritical processing so that the signals used by the critical processes are of high integrity;
- Limit the authority of signals from lower criticality levels;
- Develop an operating system that restricts the interaction between processes running on a shared resource.

4.5.3 Single Lane/Single Processor

The most basic form of system is the single lane/single processor. Such systems have a limited number of interfaces with other systems. Failure absorption will be limited, restricted to monitoring and reversionary modes of operation. Their processing power will be limited to the power of a single processor and associated interface components. Provided they have adequate resources, they will be relatively easy to develop. Also, redundancy management will be limited and there will be no major concerns related to the scheduling of tasks and the timing of data transfers.

4.5.4 Multiple Lanes/Single Processor Per Lane

The next level of system will have multiple lanes to absorb failures. Until recently, this was the most common form of system used to implement flight-critical systems such as flight control and engine management; the lanes are almost identical.

The design for a multilane system has extra design requirements; the lanes must be sufficiently separate to prevent the propagation of failures but must work together to form a coherent system. Communication between lanes is necessary:

- To align the timing of computations,
- To equalize signals that would otherwise diverge,
- To form valid signals in the presence of faults,
- To compare outputs so that faulty outputs can be isolated.

These communications are normally implemented by special links between the lanes. To reduce the number of physical links, the communication has traditionally been serial; however, as the quantity of data transferred between lanes increases, it may be necessary to implement parallel links.

Usually, the multilane/single processor per lane systems are implemented by fast general-purpose microprocessors with interconnecting links implemented by ASICs to achieve low-cost, medium-performance serial links.

4.5.5 Multiple Lane/Multiple Processor

When the throughput requirement of individual lanes is greater than a single processor can supply, or when the need for visible functional partitioning can best be achieved by physically separate processors, then each lane will have two or more processors. This provides the facilities to partition the system more clearly, but it also introduces extra requirements to ensure that communications inside a lane are coherent. Most systems allocate tasks to processors at design time and use fixed schedules to organize the timing of communications and activation of processes. Even with such systems, there are overheads associated with interprocessor communication.

More advanced systems have operating systems that allocate tasks dynamically at execution time. In such systems, an extra level of analysis is required to ensure that all tasks are achieved in a timely manner.

Most integrated vehicle management systems will be multilane/multiprocessor that will require a sound design method to ensure that communications occur at the correct times and that all tasks are completed at the required iteration rate and with the required latency. Strong hardware and software partitioning will be needed to retain the separation required to limit fault/error propagation.

4.5.6 Shared Resources

The shared resource architecture has the potential to minimize life-cycle costs. It is based on the premise that only two lanes of computation are required at any one time: one lane to provide the function and a second to check that it is producing the correct outputs. Should an error be detected, another pair of lanes is brought on line. If the same physical resources can be used for a range of tasks, it should be possible to share the standby resources between tasks.

This concept has become more viable as high-performance modules have been developed that are capable of implementing several types of functions. The reduction in design costs and spares holding has a significant effect on life-cycle costs.

However, there are disadvantages:

- The design specification has to match the requirements of the most demanding function, thus leading to high-cost modules;
- The overheads associated with task switching are significant.

If a resource is to be switched on line with high confidence that it will operate correctly, it must have been tested shortly before the switching. It must have been monitored using techniques that are independent of a specific application so that it can be used for any of the tasks. Traditionally, flight-critical systems use cross-lane monitoring based on the application; self-monitoring is used, with lower coverage, to supplement the application-based cross-lane monitoring or for maintenance functions.

The physical components of the shared resource systems are sets of common modules (processors, memory, bus interfaces, power supplies, linear interfaces, etc.) interconnected by a backplane bus and controlled by a distributed operating system.

4.6 FAULT TOLERANCE

Fault tolerance is the ability of equipment to provide its function and to continue operation after one or more faults have occurred. To provide fault tolerance, faults must be detected, identified, and isolated; redundant system resources must be available and be reconfigured to provide continuing operation.

Monitors, voters, and switching mechanisms are required to recognize faults and to provide reconfiguration paths. The recovery mechanisms must be autonomous, allowing graceful degradation in special cases. All failures that might lead to a hazard must be detected.

4.7 HARDWARE ARCHITECTURE

4.7.1 Introduction

Advances in microelectronics, fault tolerance, and software have enabled the avionics industry to develop new design concepts that result in highly integrated digital avionics under software control.

Conventional avionics consists of subsystems representing "classical" functions such as navigation, communication, and identification. These subsystems are built from line-replaceable units (LRUs), which can be replaced on the aircraft. The lowest level of unit is the shop-replaceable units (SRU), which is changed in the electronic repair shop. The systems are engineered as stand-alone systems with generic applications not foreseen.

The generic application of components is a main target in integrated avionic concepts and leads to:

- Building modular functions,
- Standardization of modules,
- Hierarchical partition and connection of modules.

Modularization creates logical, functional units with simple and well-defined interfaces. Vehicle management functions have to be defined in such a way that all functions can be realized with a minimum number of building blocks. The generic hardware building block, the hardware module, carries out different functions by running different software tasks in the same module. After a failure, the hierarchical connection of the modules permits a redistribution of functions onto the remaining healthy available resources, leading to fault-tolerant structures.

Regardless of the level of standardization, there is a need for some customization to fulfill either the desires of airlines or for military applications. This is a design driver for open, easily adaptable systems.

4.7.2 Integrated Modular Architectures

The integrated modular approach promotes the functional sharing of sensors, actuators, displays, and processors. In general, the hardware architecture consists of standardized processors communicating with each other and the other system elements (such as sensors, weapons, controls, displays, and the crew). It must satisfy the required information flows, the switching interconnect scheme, and the processing and controls.

For modularity reasons, the following basic building blocks, called line-replaceable modules (LRMs), should be available:

- Data processing element,
- Signal processing element,
- Image processing element,
- Graphics processing element,
- Mass memory element.

Power supply elements are required, either as internal functional parts of LRMs or as separate LRMs.

4.7.3 LRM Characteristics

Current LRMs have the following characteristics:

4.7.3.1 Data Processing LRM

- General data processing,
- Processing performance approximately 40–1000 MIPS,
- Structure,
  - Pipeline,
  - Parallel,
  - Dedicated cache-memory.

4.7.3.2 Signal Processing

- Processing rate approximately 100–1000 Mbit
- Functions:
  - Complex multiply/add
  - Vector/matrix operations
  - Trig functions
  - FFT, FFT^{-1} (one-dimensional)
  - FIR-Filter
  - IIR-Filter
  - Thresholding
  - Correlation
  - Modulation

4.7.3.3 Image Processing LRM

They possess the same functions as signal processing but for a two-dimensional image area.

4.7.3.4 Graphic Processing LRM

- Real-time symbol generation,
- Video distribution and display control,
- Resolution/display size,
- Three-dimensional graphics,
- Color.

4.7.3.5 Mass Memory LRM

- Approximately 100 Mbyte volatile and 10 Mbyte non volatile memory,
- Error detection and correction facilities,
- Complex address modes, logical to physical address translation.

4.7.4 Communication Network

Communication requirements for integrated modular systems call for a much higher data traffic density and number of participants than traditional systems. The data must be handled at the required priority to ensure safe, reliable operations. It comprises all layers of communication as they are defined in the ISO/OSI model [4.3].

To maintain the ability of the architecture to make standardization practical, the implementation technologies must support flexibility and growth capabilities. The communication network consists of different communication elements:

- Global links,
- Local links,
- Link control elements,
- Test/maintenance and reconfiguration bus,
- Network interconnection (bridges, routers, gateways).

4.7.5 Bus Systems

4.7.5.1 Global Buses

For global buses and civil avionics application, the ARINC 629 is utilized [4.4]. ARINC 429 can be used to transmit from point to point [4.5]. For military application, two types of buses are in use:

a) STANAG 3910 Time Division Multiplex Bus [4.13] derived from 1553-B with a transfer capacity of 20 Mbits/s.

b) High Speed Data Bus (HSDB) linear token bus [4.14] with a transfer capacity of 50 Mbit/s.

4.7.5.2 Intermodule Communication

The PI-bus [4.15] has been developed for intermodule communication (high speed, low latency). The main characteristics of the PI-bus are:

- It is linear;
- It is multisource/ multisink;
- It is synchronous (12.5 MHz);
- It supports message communication between up to 32 modules on a single backplane;
- It uses the master-slave communication protocol;
- The interface is dual independent.

The T&R-Bus provides systemwide features for test, maintenance, and reconfiguration.

4.8 SOFTWARE DESIGN/ARCHITECTURE

Software development is one of the most complex engineering activities. During the past two decades, software programs have increased in size from a few hundreds of lines of code to hundreds of thousands of lines. The software system of an IVMS is complex and large. It involves the functionality implemented, the number of variables, the amount of processing, and the level of confidence and quality that the final product must achieve. To cope with these problems, methodologies and tools have been produced, reshaped, and improved; others are now becoming available.

Software elements cover the spectrum from mission-oriented to mission-critical to safety-critical. These elements are integrated within one processing system and are required to be separated from each other to allow differing levels of verification to be applied. This separation, partitioning, or segregation will be accomplished through a combination of design processes, system architecture decisions, and software architecture implementations. Because of their interrelationship, two aspects of this multifaceted approach are discussed below: software design methods (as a subset of design processes) and software architecture implementations.

4.8.1 Software Design Methodologies

4.8.1.1 Structured Programming/Structured Languages

Dijkstra [4.6] showed that any program could be designed using only the three constructs of sequence, iteration, and selection. Since then, it has been found that constraining the number of structures in a program leads to clearer design and ultimately fewer errors. These constraints have been built into almost all the tools and languages we use.

4.8.1.2 Modular Programming/Top-Down Design

Constantine, Yourdan [4.7], and Parnas [4.8] led the way to the development of modularity as a basic feature of every software design. Even 20 years ago, high cohesion, low coupling, and data hiding were characteristics recognized as contributing to robust programs. Distinct, manageable program units could each be developed to meet their specifications and then linked through rigorous interface design. The top-down design of modular programs such as Yourdon provides a means of logically constructing a large program and has the advantage of assuring completeness [4.11, 4.12].

Since then, various theories about the rules for defining modules and their interfaces have been implemented in a number of development methodologies (see Table 4-2).

Table 4-2. Examples of Software Development Methods

MASCOT (Modular Approach to Software Construction Operation and Test) is:

- A formal method of describing the software structure of a real time system that emphasizes the communication between activities. This communication is implemented through intercommunication data area (ida). There are two types of idas:
  - Imposes a disciplined approach to yield a modular structure.
  - Provides an executive for the dynamic control of program execution at run time.

JACKSON SYSTEM DEVELOPMENT (JSD) (SPEEDBUILDER, PDF)

- JSD covers all stages of development: definition, design, and production.
- It models real-world processes to produce system functions.
- It turns specifications into realizable software, including taking into account the timing.

JSD emphasizes modeling prior to functional decomposition, coordinates processes formally, and defers the definition of the implementation.

YOURDON (TEAMWORK, SOFTWARE THROUGH PICTURES)

Yourdon is a structured design method used to create functional specifications. It creates data flow diagrams that define processes, the data flows between the processes, and the control flows that activate the processes. The control flows are triggered by the outputs from finite-state machines that are responding to input control signals.

Yourdon provides a method that uses hierarchies of processes and control that can capture complex designs systematically.

4.8.1.3 Object-Based Design

Today's networks of distributed computing nodes demand both strong independence and efficient intercommunication. Nodes have well-defined inputs and outputs; they perform precisely defined operations on their inputs; and they can be arranged in hierarchies to form larger machines out of smaller ones. Even in a single-processor multitasking system, different tasks must be encapsulated. The shift from procedure-oriented to object-oriented design supports the generation of programs with these characteristics. The key features of object-oriented design, namely, encapsulation, information hiding, and problem space orientation, lead to enhanced reuse potential, increased extensibility, and increased maintainability.

4.8.1.3.1 Objectives of Object-Orientated Implementation—

- Information hiding—A technique for encapsulating software design decisions in modules in such a way that the module's interfaces reveal as little as possible about the module's inner workings. Thus, each module is a "black box" to the other modules in the system. The discipline of information hiding forbids use of information about a module that is not in the module's interface specification [4.10].

- Problem space orientation—Organizes system objects around real-world objects such as external environment entities, hardware components, and user operations. This orientation allows the user to better understand the relationship between modules of an implementation and the real-world functions addressed by the modules.

- Encapsulation—A technique for isolating a system function within a module and providing precise specification for the module. Encapsulation groups related functions and data so that they can be treated and thought of as a unit. Encapsulation is very closely related to information hiding.

4.8.1.3.2 Advantages of Object-Oriented Design—

- Enhanced reuse potential—An object-based orientation groups data and functions cohesively. A proper balance of encapsulation and information hiding will result in objects that are units with well-defined interfaces for understanding the exact nature for interfacing with the object. Problem space orientation increases the understanding of how the software can be reused in future systems. Thus, problem space orientation makes it easier to map from future requirements to existing implementations that address the requirements.
- Increased extensibility—An object-based orientation establishes objects that are cohesive with respect to function and data, with well-defined interfaces for defining the objects. The cohesiveness and the strict interfaces make objects easier to understand, which in turn makes them easier to change. This ease of understanding is also enhanced by the problem space orientation because the mapping from the problem space to the implementation is easier to make. Proper encapsulation and information hiding mean that changes to particular objects are confined within the particular object. Only when changes affect an object's interface is it necessary to investigate the impact of the changes on the rest of the system.
- Increased maintainability—An object-based orientation increases maintainability by establishing objects with well-defined interfaces for easier localization of change. As with extensibility, the objects are also easier to understand, which is vital when trying to maintain a system long after its original development phase. Taking a problem space orientation also makes it easier to map future enhancements to existing objects that are affected.

4.8.1.4 Diversity

Diversity is a technique used to provide protection from errors created during the development of algorithms and/or software.

If two diverse versions of an algorithm or a software program are developed, it is possible to cross-check the outputs from the diverse programs and detect errors. When an error is detected, the system is commanded to a safe state.

If more than two versions of an algorithm or a software program are developed, it is possible to cross-compare the outputs, reject an erroneous output, and continue to provide correct system operation.

The multiple versions of the algorithms/programs can have varying degrees of diversity.

- The algorithms/programs may be based on different physical phenomena. For example, the protection against some type of error in a nuclear plant may be based on temperature with a diverse protection provided by monitoring pressure. Some types of errors in aircraft control systems can be monitored by modeling the control system with diverse protection provided by observing the accelerations and body rates.
- Diversity may be provided by developing multiple versions of a program to meet the same requirements specification (n-version programming).

Diversity is provided by a combination of the following techniques:

- Using separate teams to develop the programs;
- Using different languages for each program;
- Using different target hardware for each program;
- Using a common language but different compilers, different run time systems, and different target hardware.

However, to obtain these benefits, one has to pay for the development of n versions and the associated expense of the extra final testing of the integrated versions.

4.8.1.5 Formal Methods

Formal methods of software development use mathematical notations, models, or algebras to specify systems and algorithms and convert the algorithms into an imbedded software program. Specifications written in formal languages such as Z, VDM, or B can be proved to have certain key properties. The process of converting the high-level specification to code can also be proved to be correct.

Structured programming has been used successfully for many years to improve the quality of software products, eliminating error sources and improving testability. However, today it is necessary in some systems to use set theory and logic to describe system specifications and software designs with a more mathematically formal language. The UK Interim Defense Standard 00-55 has made formal methods virtually mandatory for certain classes of safety-critical software [4.9].

If a language can be defined that is more precise and universal than natural languages, it must be used as much as possible in the software/systems community. The mathematical tools used to support formal methods are becoming easier to understand and use.

The result should be better specifications and programming, fewer errors, and improved efficiency in the process and in the product. The verification, validation, and testing workload should also be reduced considerably, leading to cost savings. Affordability will be improved, and complex large systems will become less costly and technically more feasible.

4.8.2 Safety-Critical Software Architecture

The architecture of a software system for safety-critical functions must ensure that the very high reliability requirements are met. It must provide protection from the weaknesses of software within the system that satisfy less stringent reliability requirements. While achieving the benefits of integration of vehicle management functions, current safety-critical-only software architectures must be modified to be compatible with software developed to lower standards.

4.8.2.1 Software Architectures for Safety-Critical Functions

The architecture for safety-critical software has the following characteristics:

- Simple hierarchical structures to aid comprehension;
- Modular to facilitate a gradual buildup of verification;
- Simple constructs to aid verification;
- Uses strong typing to reveal typing errors;
- Uses real-time schedules to avoid the problems of concurrency;
- Maps to the functions being implemented to minimize the impact of change and to increase the predictability of the effect of failures and errors;
- Basic form that is:
  - Process input signals,
  - Compute function,
  - Format output signals,
  - Include one or more voting plains to detect and isolate failures,
  - Separate in-flight functions from maintenance functions.

4.8.2.2 Software Architectures for Embedding Safety-Critical Functions in an Integrated Vehicle Management System

Early software development efforts associated with the control and delivery of nuclear weapons and, later, commercial nuclear systems addressed the issue of assuring that related or simple cohabiting functions did not have any possibility of overcoming the safeguards built into the primary software. More recently, the very different considerations of personal privacy and national data security are leading to very rigid rules for software and hardware architecture construction. These considerations demand that rules of access be embedded in the software. This, in turn, requires that the rules themselves be protected against either deliberate or inadvertent alteration. (It is the latter that concerns our application.) In the NSA world, the concept of a trusted computing base (TCB) has evolved. At the centre of a TCB is a trusted kernel.

4.8.3 Development Phasing

Phasing the software development process and partitioning the software is necessary to handle complexity. Such phasing and partitioning permit a group of staff to work on the same product without increasing the number of errors. The traditional phases are: requirements analysis, system specification, system design, coding, module testing, integration testing, acceptance, and maintenance.

The traditional method of partitioning software is to split it into modules loosely based on function.

4.8.4 Verification of Software

Verification during all processes is essential to improve the correctness of transmitting information from phase to phase and person to person.

Verification consists of testing, inspection, and analysis. In the early phases, verification is mainly by inspection with reviews and analysis; later phases are mainly concerned with testing and the analysis of coverage that the testing achieves.

Test and fault removal are an important and expensive task in all the required processes. They have to be applied at module, program, integrated hardware-software, rig, and aircraft levels.

It ensures the early removal of all detected errors and leads to a better final system.

4.9 COMMERCIAL-OFF-THE-SHELF TECHNOLOGY

A new and large area of untapped cost savings concerns joint application of common commercial and military electronic parts and systems. While entire systems can be examined, cockpit displays, for example, the use of commercial parts and practices in designing military electronics systems can have significant payoff. The challenge is to overcome decades of methods and experiences the military had with specifications, inspection and test practices and certain technologies.

The advantages of using commercial parts in military equipment are:

- Significant cost savings,
- Declining availability of military parts,
- Improvements in plastic part technologies,
- State of the art has moved from military to commercial sectors.

Several affordable avionics definitions were given that apply to this study:

- MIL specification parts—Hermetically sealed microcircuit and passive electronic components meeting military processing and testing requirements, typically:
  - MIL-STD-883,
  - MIL-M-38510,
  - DESC drawing.
- Commercial parts—Qualified ceramic or plastic microcircuits and passive components not screened to military standards
- Commercial practice—Manufacturing processes and control systems in place in the commercial electronic industry founded on principles of:
  - Continuous improvement,
  - Statistical process control,
  - Root cause failure analysis and correction,
  - Key supplier relationship.
- MIL specification practice—Manufacturing processes and control system adhering to rigid DoD specifications defining product performance and process methodology.

  The strategic implications of using commercial parts in military equipment are:
  - Reduced cost;
  - Reduced cycle time;
  - Increased technological advancement (faster processing speeds, smaller systems, etc.);
  - More efficient quality practices.

  The barriers to the use of commercial parts are:
- Military procurement practices,
- Technical and/or cultural reasons,
- Political,
- Self-imposed,
- Contract parts selection requirements,
- MIL-HDBK-217 does not reflect correct real of relative performance data,
- Nonstandard part approval cycle is difficult and expensive,
- Initial implementation cost.

The good news, however, is:

- There is excellent experience with commercial parts in the commercial aircraft sector;
- Some DoD programs are allowing selected use of commercial parts;
- Supporting data and studies are accelerating affordable avionics;
- Army/CALCE Physics of Failure Research;
- Stress Margin Approach (SMA);
- DoD Microcircuit Planning Group (DMPG);

- Defense Logistics Agency (DLA) Mantech Plastic Packaging Availability Program;
- ASAF Wright Laboratory's Reliability without Hermeticity (RwoH);
- ASAF Electronic System Division (ESD) Commercial Component Initiative;
- A long-term warranty approach to ensuring design integrity exists.

To illustrate the differences in practices and technologies, Figure 4-5 shows the reliability of electronics developed by military versus commercial requirements. The case described contains a number of development distinctions:

- The hermetic curve illustrates military practices and specifications:
  - Rigid Mil Spec inspection of all parts,
  - Designed for high reliability,
  - Expensive manufacturing,
  - High cost per part due to low volumes,
  - Few suppliers.
- The nonhermetic curve refers to plastic parts using commercial practices and procedures:
  - Commercial statistical process control,
  - Designed for high reliability,
  - Low-cost manufacturing,
  - Low cost per part due to high volumes,
  - Many suppliers to guarantee availability.

Remarkably, over time, the commercial products have proven to be as reliable as the military products.

In recognition of this kind of performance, U.S. Secretary of Defense William Perry announced in 1994 a bold new initiative concerning the use of commercial parts and practices. The following statement is part of this initiative:

> ***Military specs and standards:*** *Performance specs shall be used when purchasing new systems, major modifications, upgrades to current systems, and non-developmental and commercial items for programs in any acquisition category, ...the use of mil-specs and standards is authorized as a last resort with an appropriate waiver.*

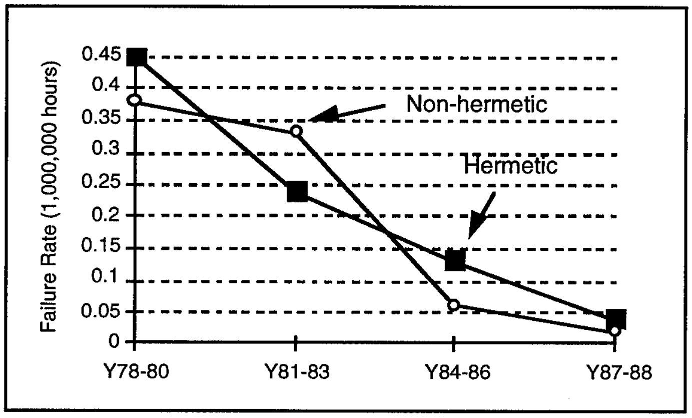


Figure 4-5. Military versus Commercial Parts

4.10 PRACTICAL CONSIDERATIONS IN IVMS DESIGN

4.10.1 Development Costs

The development costs of newly designed IVMSs are very large. In many cases, the number of production aircraft that can be anticipated is insufficient to allow an affordable recovery of IVMS development costs. Thus consideration should be given to transferring intact major components of a new IVMS from previous proven designs.

4.10.2 Verification

Component or module-level testing is applied to ensure compliance with specification. Integration of these components or modules into larger physical subsystems makes it possible to perform major vehicle functions. Consideration should be given during the design phase to including means for stimulation of the system for test purposes and to providing visibility of system operation through access to system parameters.

4.10.3 Validation

The verified physical subsystems are integrated into a complete IVMS with full functionality. The complete IVMS is evaluated in a run-time configuration with full simulation of the vehicle's environment. The system's ability to perform the desired mission is assessed under normal operating conditions and under a range of failure conditions.

4.10.4 Flight Test

4.10.4.1 Component Verification Flight Test

Testing of system components, particularly sensor and actuator complexes that are more likely to be affected by the environment, are best done under controlled conditions on a testbed aircraft. High-quality instrumentation is required to characterize the test environment to the degree necessary to provide complete characterization of conditions.

4.10.4.2 Integrated System Validation Flight Test

Flight test validation is required of a representative integrated system (possibly one side of the vehicle). This should be either the full-up system or a major part of the full-up system. The requirement is to expose the system to the actual operating environment, including the system time delays, which may couple with the pilot/vehicle response to create PIO conditions.

4.10.4.3 Certification Flight Test

These tests check the full IVMS in a production vehicle to demonstrate compliance with all the certification requirements.

4.10.5 Testing Tools

The test engineer requires tools to provide high visibility of system operation during test. This includes immediate documentation of internal system behavior during response to failures, reconfiguration, and other events of interest.

4.11 CONCLUSIONS

This chapter contains a discussion of the many possible forms of vehicle management systems. The key driver for integration is cost rather than technology. Numerous architecture philosophies can be applied. Architectures that maintain simplicity without sacrificing throughput and memory are typically the best for VMS. The main concern is the mixing of critical and noncritical signals. Also, there is a need to avoid all parts of the systems ending up at the top level of criticality; hence, there is a need for architecture(s) that allow separation of critical and noncritical functions.

4.12 REFERENCES

[4.1] Tooze, M. J., "System Engineering-Avionics System Design Methodology," AGARD Lecture Series No. 164.

[4.2] Meyer, B., "Object Oriented Software Construction," Prentice Hall, 1988.

[4.3] STANAG 4254.

[4.4] Airlines Electronic Engineering Committee, ARINC Specification 629, "Multi Transmitter Data Bus, Part I: Technical Description," March 7, 1990.

[4.5] Airlines Electronic Engineering Committee, Supplement 10, "ARINC Specification 429: Mark 33 Digital Information Transfer System (DITS)," March 16, 1987.

[4.6] Dijkstra, E. W., "Selected Writings on Complexity: A Personal Perspective," Springer-Verlag, New York, 1982.

[4.7] Constantine, L. L., and Yourdan, E., "Structured Design," Prentice Hall, 1988.

[4.8] Parnas, D., Clements, P., and Weiss, D., "The Modular Structure of Complex Systems," *Proceedings, Seventh International Conference on Software Engineering*, March 1984; reprinted in *IEEE Transactions on Software Engineering*, Vol. SE-11, March 1985.

[4.9] UK Interim Defense Standard 00-55, "Hazard Analysis and Safety Classification of the Computer Programmable Electronic System Elements of Defense Equipment."

[4.10] ANSI/MIL-1815A-1983, "Ada Language Reference Manual."

[4.11] Birrell, N. D., and Ould, M. A., "A Practical Handbook for Software Development," Cambridge University Press, 1988.

[4.12] DeMarco, T., "Structured Analysis and System Specification," Yourden Press, 1978.

[4.13] STANAG 3910.

[4.14] SAE AIR-4288, "Linear Token Passing Multiplex Data User's Handbook."

[4.15] SAE AS-4710, "Pi-bus"

CHAPTER 5

EXAMPLES OF VMS DESIGNS

5.1 INTRODUCTION

The purpose of this chapter is to present some examples of modern aircraft employing VMS implementations. The examples will illustrate principles and general concepts employed in deploying an IVMS. Four examples have been chosen to cover different aerospace vehicles. These vehicles include the B-777 commercial transport, the F-22 Advanced Tactical Fighter, the RAH-66 Comanche Reconnaissance/Attack Helicopter, and the Experimental Aircraft Program (EAP).

5.2 B-777

The B-777 is the latest addition to the Boeing family of commercial aircraft. This aircraft incorporates a mix of proven equipment, many new technologies, and some new features. The all-digital aircraft contains over five million lines of code so that V&V became increasingly challenging. The B-777 architecture is shown in Figure 5-1.

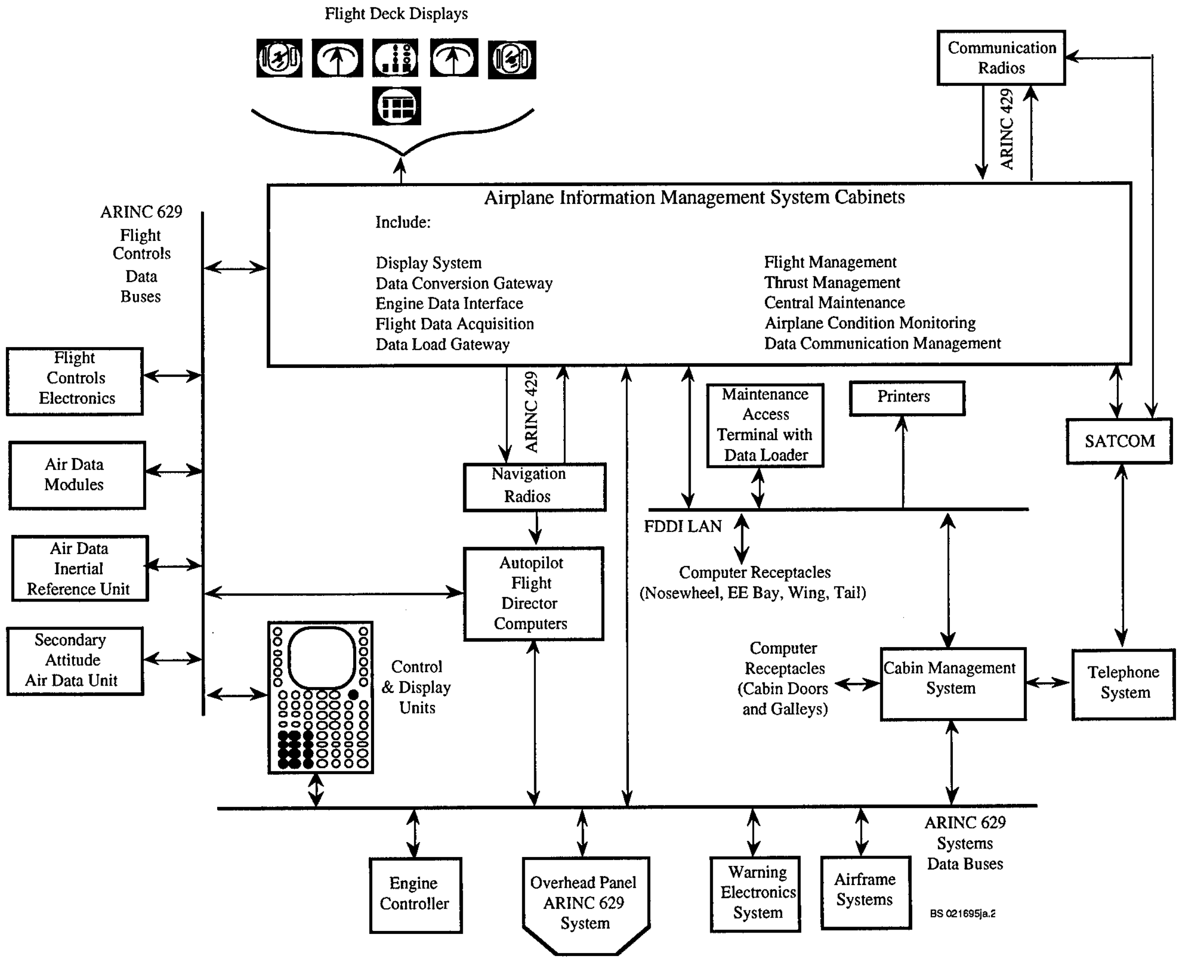
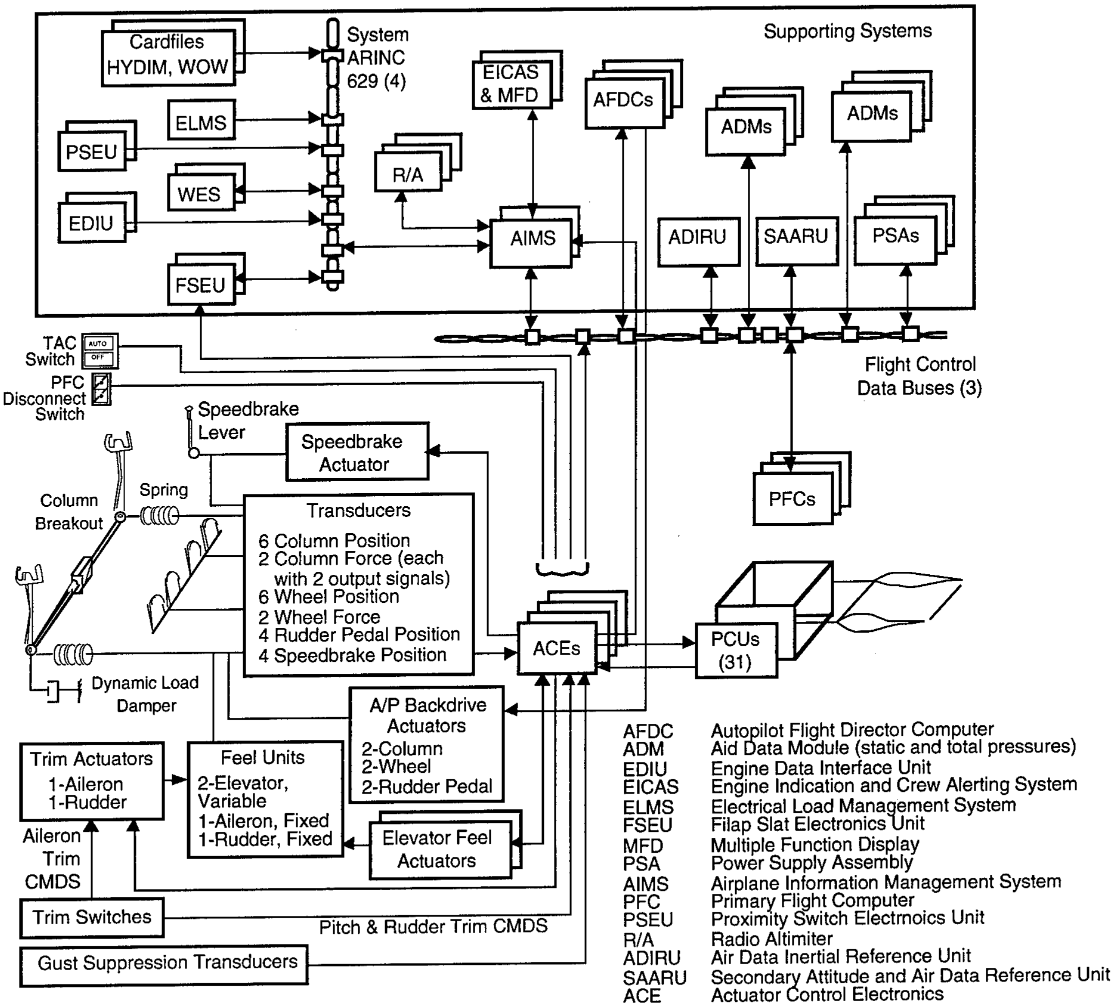

Flight Deck Displays
Communication Radios
ARINC 429
ARINC 629 Flight Controls Data Buses
Airplane Information Management System Cabinets
Include:
Display System
Data Conversion Gateway
Engine Data Interface
Flight Data Acquisition
Data Load Gateway
Flight Management
Thrust Management
Central Maintenance
Airplane Condition Monitoring
Data Communication Management
Flight Controls Electronics
ARINC 429
Maintenance Access Terminal with Data Loader
Printers
SATCOM
Air Data Modules
Navigation Radios
FDDI LAN
Air Data Inertial Reference Unit
Computer Receptacles (Nosewheel, EE Bay, Wing, Tail)
Autopilot Flight Director Computers
Secondary Attitude Air Data Unit
Control & Display Units
Computer Receptacles (Cabin Doors and Galleys)
Cabin Management System
Telephone System
ARINC 629 Systems Data Buses
Engine Controller
Overhead Panel ARINC 629 System
Warning Electronics System
Airframe Systems
BS 021695ja.2

Figure 5-1. B-777 System Architecture

The areas of primary interest to vehicle management are the flight control system and the Aircraft Information Management System (AIMS). Also, the Air Data Inertial Reference Unit (ADIRU) is a good example of functional integration.

5.2.1 Flight Control [5.11]

The B-777 Primary Flight Control System is shown in Figure 5-2 and consists of triple redundancy for all hardware resources: computers, airplane electrical power, hydraulic power, and communications. This extraordinary redundancy and dissimilarity is utilized to met extremely high levels of functional integrity and availability of the fly-by-wire (FBW) flight control system.



AFDC	Autopilot Flight Director Computer
ADM	Aid Data Module (static and total pressures)
EDIU	Engine Data Interface Unit
EICAS	Engine Indication and Crew Alerting System
ELMS	Electrical Load Management System
FSEU	Filap Slat Electronics Unit
MFD	Multiple Function Display
PSA	Power Supply Assembly
AIMS	Airplane Information Management System
PFC	Primary Flight Computer
PSEU	Proximity Switch Electmoics Unit
R/A	Radio Altimiter
ADIRU	Air Data Inertial Reference Unit
SAARU	Secondary Attitude and Air Data Reference Unit
ACE	Actuator Control Electronics
PCU	Power Control Units, Actuators
HYDIM	Hydraulic Interface Module
WOW	Weight on Wheels
WES	Warning Electronics System

C951016-41

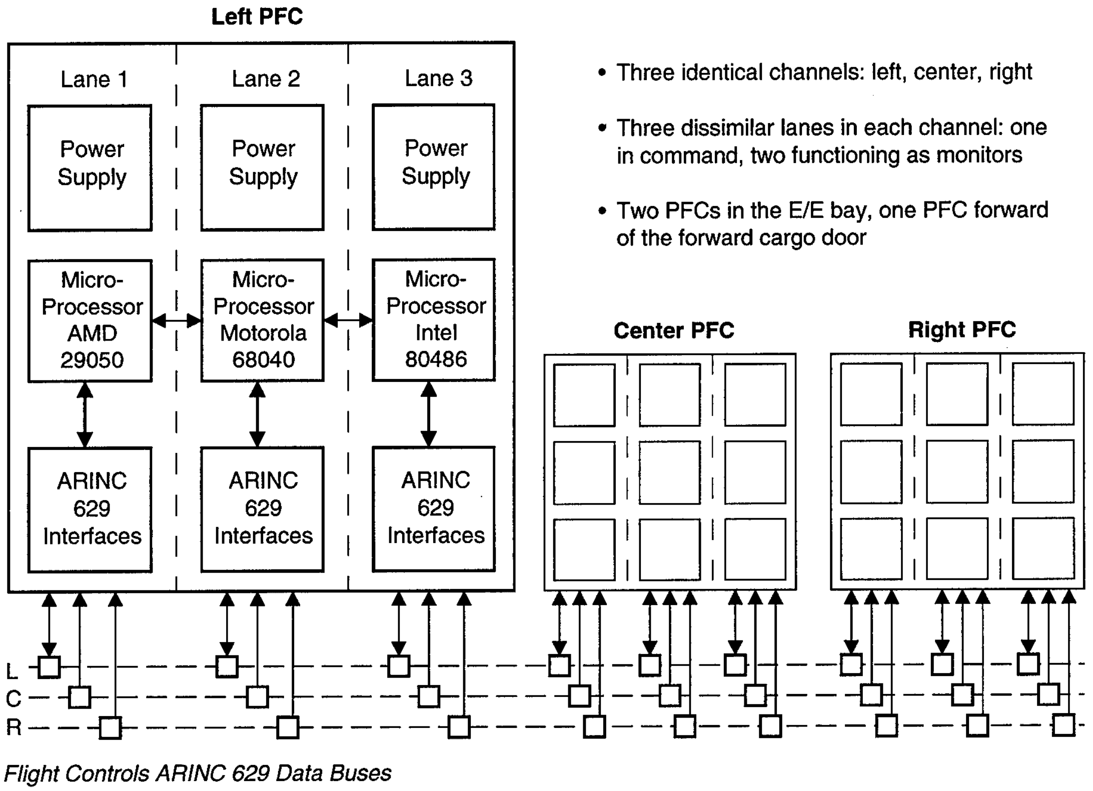
Figure 5-2. 777 Primary Flight Control System

The three elements of this design that relate to VMS are

- Primary Flight Computers (PFCs),
- Actuator Control Electronics (AEC),
- Flight Controls ARINC 629 Bus.

5.2.1.1 Primary Flight Computers (PFCs)

Shown in Figure 5-3, three PFCs provide triple-redundant computational channels for the primary flight control system. Each PFC receives data from all three flight control data buses but can transmit on only one. Within each PFC are three internal communication lanes, with each lane communicating with all three data buses using dedicated hardware. Each PFC channel contains a high degree of dissimilarity: unique microprocessor and power supply plus unique Ada source code using three different Ada compilers.



- Three identical channels: left, center, right
- Three dissimilar lanes in each channel: one in command, two functioning as monitors
- Two PFCs in the E/E bay, one PFC forward of the forward cargo door

C951016-40

Figure 5-3. Primary Flight Computer Channel Architecture

This type of massive dissimilarity results from a desire to account for and preclude "generic" failures. The existence, nature, and importance of these failures are quite controversial, particularly in light of the potentially enormous costs of creating these dissimilar structures. These concerns are raised because of the desire to make the aircraft extremely safe. Safety requirements apply to PFC failures, which could preclude continued safe flight and landing, and include both passive failures (loss of function without significant immediate airplane transient) and active failures (malfunctions with significant immediate airplane transient).

This architectural concept virtually precludes the use of common modules. Discussion of the AIMS concept in Subsection 5.2.2 deals with common modules and fault tolerance in a fundamentally different way.

5.2.1.2 Actuator Control Electronics

Each of four ACEs provides the interface between the analog and digital portions of the FBW system. This would include position transducers and servo loop electronics for all aircraft surfaces and variable feel actuators. Each ACE also contains three terminals to communicate with the ARINC 629 data buses.

5.2.1.3 Flight Controls ARINC 629 Bus

The B-777 global data bus is the ARINC-629. This new system of data buses use two-way, multitransmitter, autonomous terminal controllers. Each bus can handle up to 2 mbit/s. With the consolidation of a number of other bus types, considerable attention was paid to bus integrity and definition of protocol.

The requirements for flight control are dictated by FBW functional requirements and automatic landing.

1. Loss of one flight controls ARINC 629 bus shall be no greater than 10^{-5} per flight hour,
2. Loss of two flight controls ARINC 629 buses shall be no greater than 10^{-9} per flight hour,
3. Loss of three flight controls ARINC 629 buses shall be no greater than 10^{-11} per flight hour.

5.2.1.4 Triple Dissimilarity

As mentioned above, the 777 primary flight control system uses extensive dissimilarity to achieve coverage of generic failures. This concept, called triple dissimilarity, can be summarized as follows [5.11]:

Primary Flight Computers

- Dissimilar Processors and Compilers (Common Software),
- DO-178 Development Process,
- ASIC Development Process,
- Analysis and Testing.

Actuator Control Electronics

- Dissimilar Monitor and Control Functions,
- ASIC Development Process,
- Analysis and Testing.

Inertial Data

- Dissimilar ADIRU/SAARU,
- DO-178 Development Process,
- Analysis and Testing.

Autopilot Flight Director Computer (AFDC)

- DO-178 Development Process,
- ASIC Development Process,

- Dual Dissimilar Hardware for Backdrive Function,
- In-Service Experiences,
- Analysis and Testing,
- Limited Exposure Time for Autoland.

ARINC 629

- Development Process,
- Analysis and Testing,
- ACE Direct Mode Which Bypasses ARINC 629.

5.2.2 Aircraft Information Management System (AIMS)

The AIMS architecture is the first example of the use of modular concepts in commercial aviation. As such, the design follows the lead of the military avionics architecture for the F-22. What makes the AIMS concept even more relevant to VMS is the use of advanced fault tolerance concepts to implement this commercial flight-critical system (10^{-9} probability of catastrophic failure per hour).

The AIMS architecture

- Incorporates open commercial networks, buses, and point-to-point links.
- Incorporates modular cabinetry within the equipment bay that embodies Integrated Modular Avionics (IMA) concepts. Similar to the objectives of the military JIAWG, PAVE PILLAR, and PAVE PACE activities, the airlines have developed IMA to gain the advantages of:
  - Higher quality at service entry;
  - Higher MTBUR, MTBUR/MTBF ratio (fewer "NO FAULT FOUNDS");
  - Improved fault isolation;
  - Improved dispatch reliability;
  - Deferred maintenance capability;
  - Reduced training maintenance manuals;
  - Improved maintenance diagnostics repair time;
  - Reduced test equipment requirements;
  - Reduced spares requirement;
  - Reduced weight;
  - Reduced spares cost;
  - General-purpose architecture that supports in-service changes;
  - Functional expansion within cabinet versus additional LRUs.
- Incorporates an airplane Central Maintenance Computer (CMC) that provides integrated, onboard maintenance diagnostics for the airplane.
- Provides the following IMA avionics integration values to the airlines:
  - Cost per function improvement: 70%;
  - Reliability improvement: 80%;

- MTBUR/MTBF ratio goal: 90%;
- Dispatch reliability target: 99%.

5.2.2.1 ARINC 659 SAFEbus

The backbone of the commercial IMA approach is the ARINC 659 SAFEbus backplane [5.2]. This highly fault-tolerant, high-bandwidth bus is founded on technologies developed by the military for modular flight controls and adopted by the B-777 program to accommodate the objectives mentioned above. Key attributes include the notion of simple hardware self-checking pairs at the bus interface. Shown in Figure 5-4, this concept was developed for U.S. Air Force fault-tolerant flight control applications in the mid 1980s.

The SAFEbus is an enabling technology and is discussed in detail in Subsection 6.4.4.3.

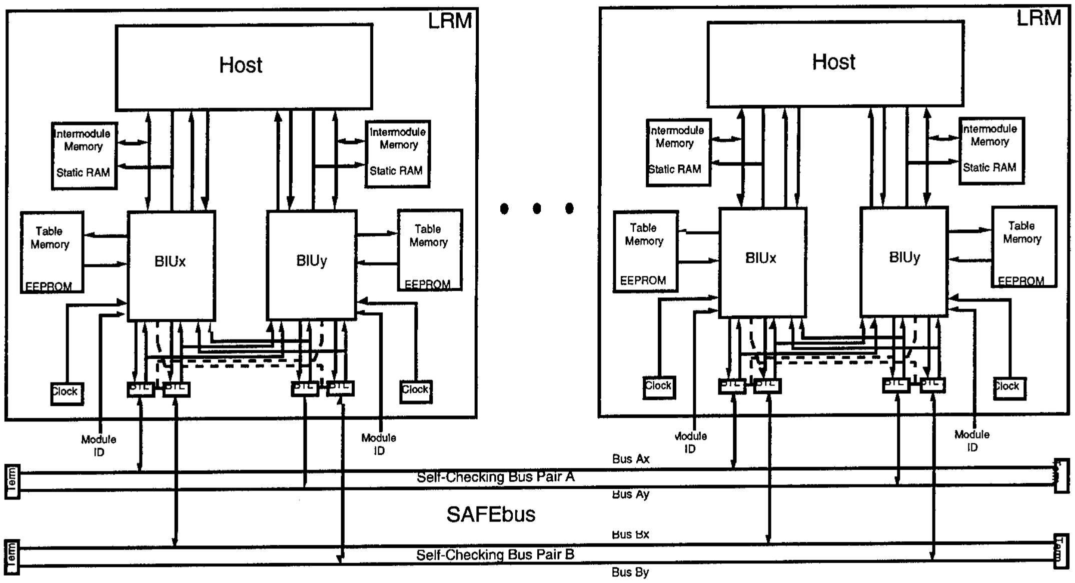


Figure 5-4. B-777 SAFEbus Implementation

5.3 F-22 ADVANCED TACTICAL FIGHTER

The ATF program objectives are to increase capabilities and reduce LCC through the use of an integrated system approach. These objectives are being achieved, in part, through the use of avionics methods that are modular, fault tolerant, and reconfigurable. Integrated systems permit a higher level of functional integration, which leads to increased operational capability. Integration also allows the application of standard hardware and software modules, which contributes to reduced LCC. The application of fault-tolerant and reconfigurable techniques substantially increases system reliability, further enhancing LCC and mission capability. The program proceeded from a prototype phase (YF-22 and YF-23) into the EMD phase. The YF-22 will be described below.

5.3.1 YF-22 Prototype

The prototype phase was a risk-reduction effort with the VMS system integrated with off-the-shelf hardware. This allowed demonstration of capabilities, evaluation of functionalities, provision of an experience base for the development team, and provided time for the development of the avionic hardware. Figure 5-5 illustrates the YF-22 configuration. As no mission capability was demonstrated on this vehicle, the avionics bus was single channel and

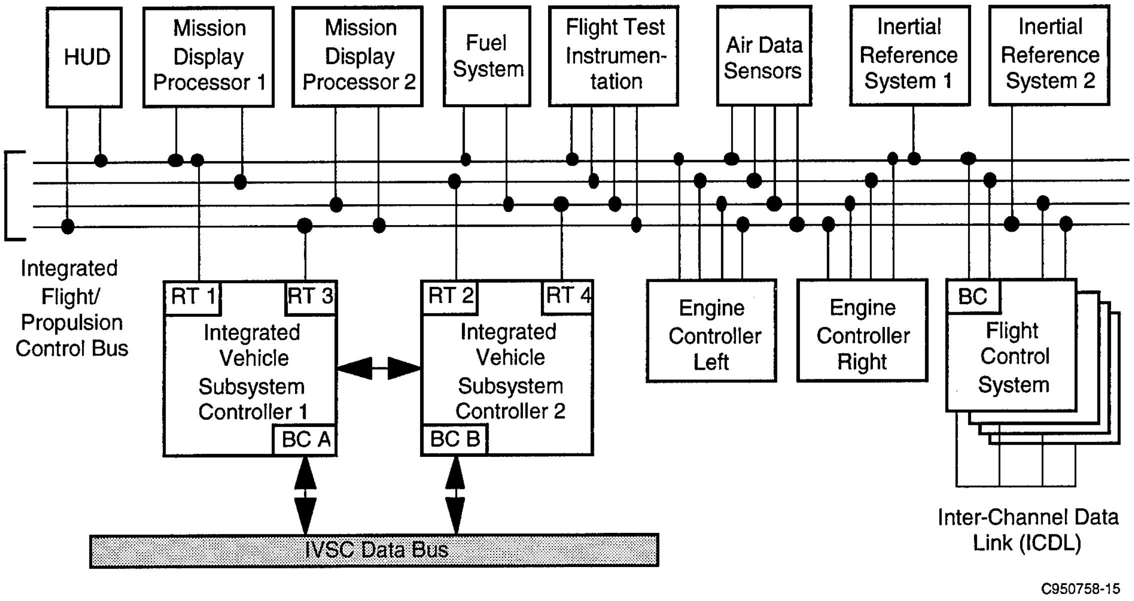

HUD
Mission Display Processor 1
Mission Display Processor 2
Fuel System
Flight Test Instrumentation
Air Data Sensors
Inertial Reference System 1
Inertial Reference System 2
Integrated Flight/ Propulsion Control Bus
RT 1
RT 3
Integrated Vehicle Subsystem Controller 1
BC A
RT 2
RT 4
Integrated Vehicle Subsystem Controller 2
BC B
Engine Controller Left
Engine Controller Right
BC
Flight Control System
Inter-Channel Data Link (ICDL)
IVSC Data Bus
C950758-15

Figure 5-5. YF-22 Prototype Architecture

coupled inertial data and displays. The VMS consisted of a quad 1553 IFPC data bus that physically integrated all flight-critical systems: quad flight control, vehicle state sensors, engine controllers (two versions triplex and quad), IVSC controllers, fuel system, controller, and displays. The IVSC Bus coupled the utility subsystem, which consisted of: electrical, hydraulic, environmental control, landing gear, life support, auxiliary power, and fire protection. Several integrated control functions were demonstrated and will be described in the F-22 section.

5.3.2 F-22 Avionics Architecture

The overall F-22 architecture is illustrated in Figure 5-6. The features of this mission or payload system are:

- High-bandwidth optical point-to-point data buses to link sensors with signal processors;
- Common data and signal processing modules for reduced cost of ownership and provision of high-performance building blocks;
- Advanced cockpit with flat-panel color displays;
- Coupling with VMS;
- Integrated CNI radio functions;
- Ada software.

The F-22 interconnect network consists of a dual 1553 data bus as shown in Figure 5-7. The bus physically couples the VMS, IVSC, and engine diagnostics with the IRS, SMS, displays, and central processors. As noted in the diagram, the VMS has responsibility for primary bus control, with secondary control passing to the IVSCs.

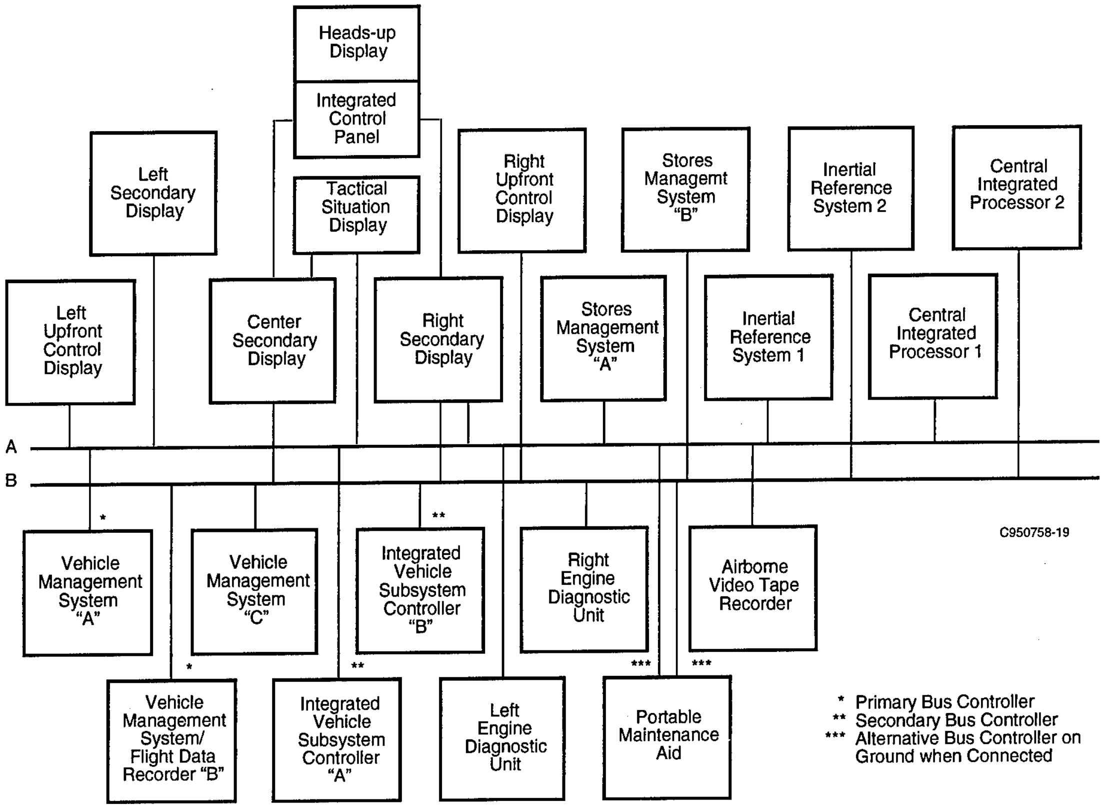

Heads-up Display
Integrated Control Panel
Tactical Situation Display
Left Secondary Display
Right Upfront Control Display
Stores Managemt System "B"
Inertial Reference System 2
Central Integrated Processor 2
Left Upfront Control Display
Center Secondary Display
Right Secondary Display
Stores Management System "A"
Inertial Reference System 1
Central Integrated Processor 1
A
B
*
**
C950758-19
Vehicle Management System "A"
Vehicle Management System "C"
Integrated Vehicle Subsystem Controller "B"
Right Engine Diagnostic Unit
Airborne Video Tape Recorder
*
**
***
***
Vehicle Management System/ Flight Data Recorder "B"
Integrated Vehicle Subsystem Controller "A"
Left Engine Diagnostic Unit
Portable Maintenance Aid
* Primary Bus Controller
** Secondary Bus Controller
*** Alternative Bus Controller on Ground when Connected

Figure 5-6. F-22 Avionics Bus

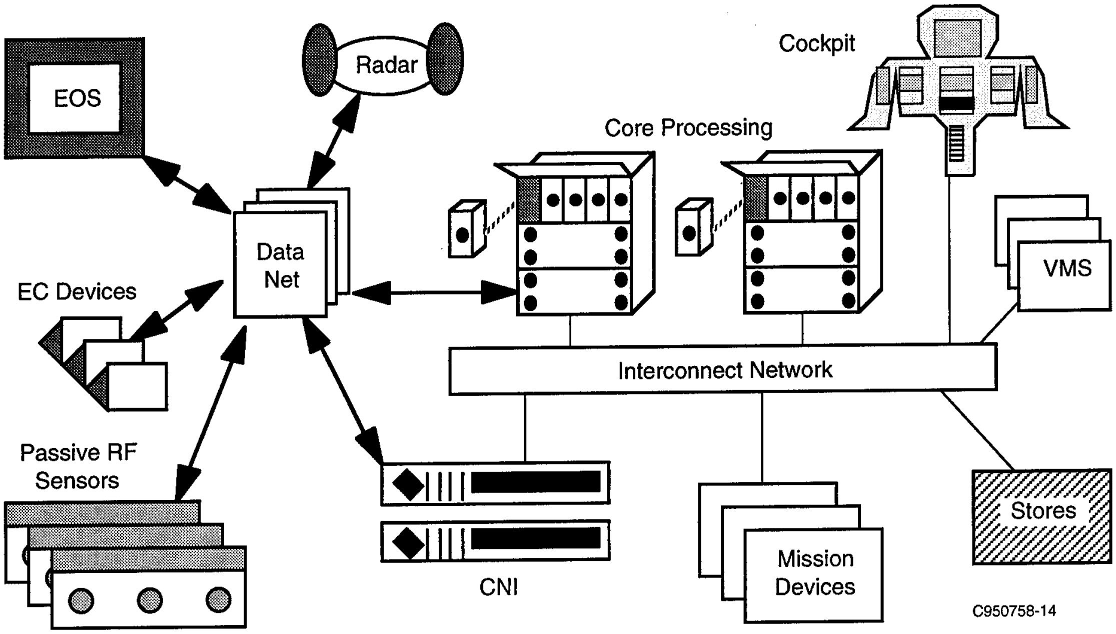


Figure 5-7. F-22 Avionics Architecture

5.3.3 F-22 VMS

In the F-22, the VMS is defined as the IFPC system. Other subsystems such as electrical power, environmental cooling, landing gear, and so on, are considered part of the IVSC. In the context of this report, the combined VMS and IVSC form the basis of an IVMS.

The F-22 architecture is shown in Figure 5-8. The system consists of the IFPC and IVSC global buses, an interface to the avionics system bus, the PI and I/O local buses, and an interface to the FADECs.

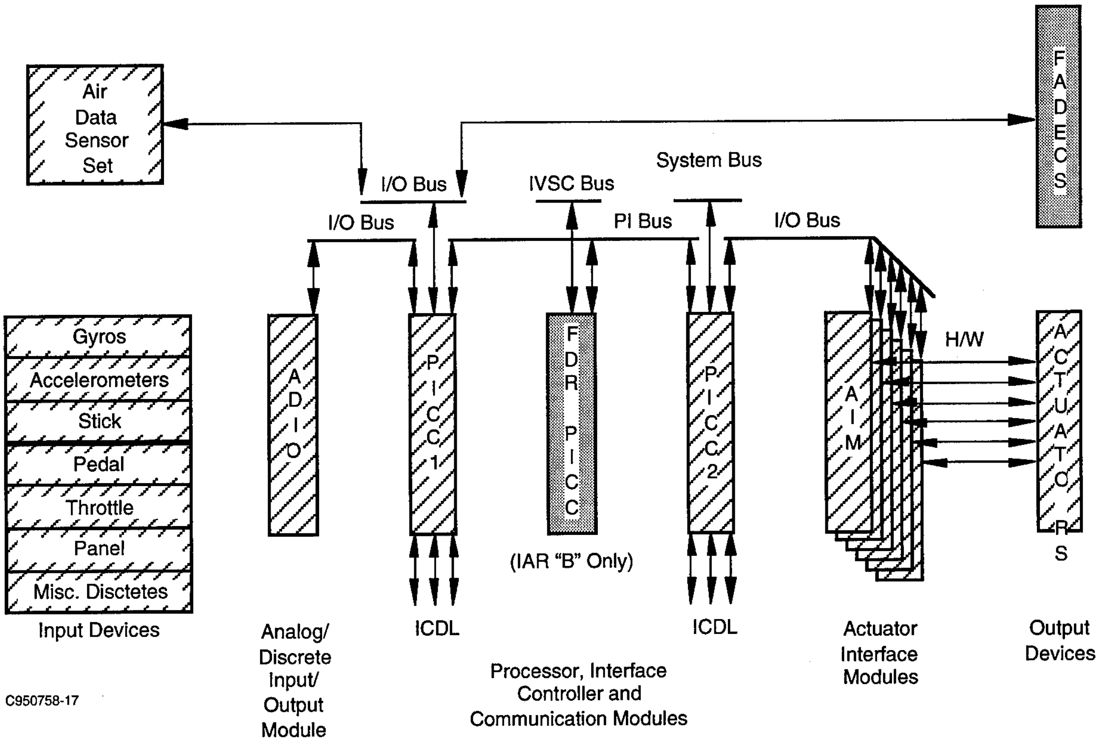


Figure 5-8. F-22 VMS Architecture

PICC Module 1 contains the software for the control law outer loop functions, air data computations, redundancy management, and BIT. PICC Module 2 contains the software for the actuator monitors, output monitors, redundancy management, and BIT. The utility systems are controlled and monitored by software in the IVSC.

The major elements of the system will be described in more detail below.

5.3.3.1 Bus Types

As is shown in Figure 5-9, three bus types are utilized in the VMS. The global bus is the IFPC 1553 bus, which couples air data sensing, engine controllers, and processing. The PI backplane bus couples the processing modules in the VMS rack, and the I/O bus handles data transfers from the A/D sensor input modules and between-processor output to the actuator interface module. Although not shown, the IVSC global data bus is also a 1553 bus and couples the various subsystem controllers. Figure 5-9 shows the IFPC bus architecture.

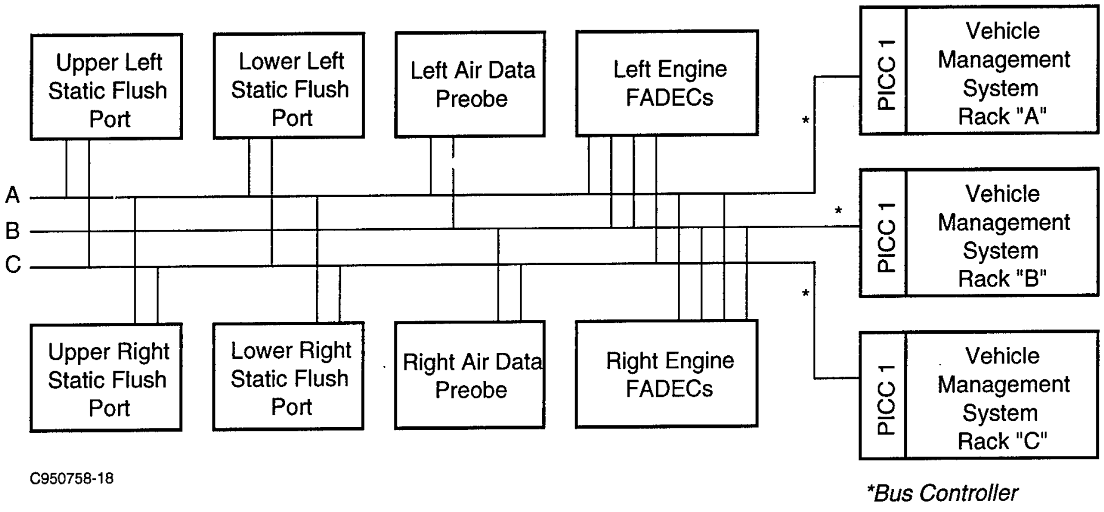

Upper Left Static Flush Port
Lower Left Static Flush Port
Left Air Data Preobe
Left Engine FADECs
PICC 1
Vehicle Management System Rack "A"
A
B
C
PICC 1
Vehicle Management System Rack "B"
PICC 1
Vehicle Management System Rack "C"
Upper Right Static Flush Port
Lower Right Static Flush Port
Right Air Data Preobe
Right Engine FADECs
C950758-18
*Bus Controller

Figure 5-9. IFPC Bus Architecture

5.3.3.2 Interfaces

The major interfaces of the VMS are to the FADEC through the 1553 data bus. The VMS interface enables the integrated flight/propulsion control functions. The other 1553 bus interface is to the air data sensor set. Vehicle state sensors and pilot controls are hardwired into the ADIO, with the digital outputs of the ADIO passed to PICC over an I/O bus. Actuator commands are passed from PICC 2 to the AIMs over an I/O bus, with the outputs hardwired to the actuators. Interface to the IVSC is through the FDR PICC and to the avionics buses through PICC 2.

5.3.3.3 Sensor/Actuator Types

The hardware arrangement within the F-22 VMS is shown in Figure 5-10. Direct-drive valves (DDV) are used on the flight-critical surfaces (horizontal tails, flaperons, and leading-edge flaps) and are supplied by two hydraulic supplies. Electrohydraulic valves (EHV) are used for the remaining surfaces and subsystems (ailerons, rudders, nose wheel steering, bypass doors) and are supplied with one hydraulic supply. The vehicle dynamic state is sensed by rate gyros and accelerometer assemblies. The ADS consists of left and right air data probes and four flush-mounted ports.

5.3.3.4 Redundancy

The F-22 FLCS is triplex redundant without either a hardware or software backup mode. The system is synchronized to allow for failure isolation between channels when driving a DDV. The IVSC is a duplex system. The propulsion control system is quad in a dual-dual configuration.

5.3.3.5 Avionics Modules

Table 5-1 lists the features of the VMS modules that were depicted in the VMS architecture, Figure 5-8. As noted, four types of modules are utilized in the IAR per IFPC channel. It is observed that the PICC has the broadest usage, followed by the PS module. As can be seen, the ADIO and AIM are flight control specific and would have the lowest utilization. The modules are packaged in a 1/2 ATR size rack, as shown in Figure 5-11.

There is one rack per channel. The FADECs are engine mounted and mechanized with 1750A processors.

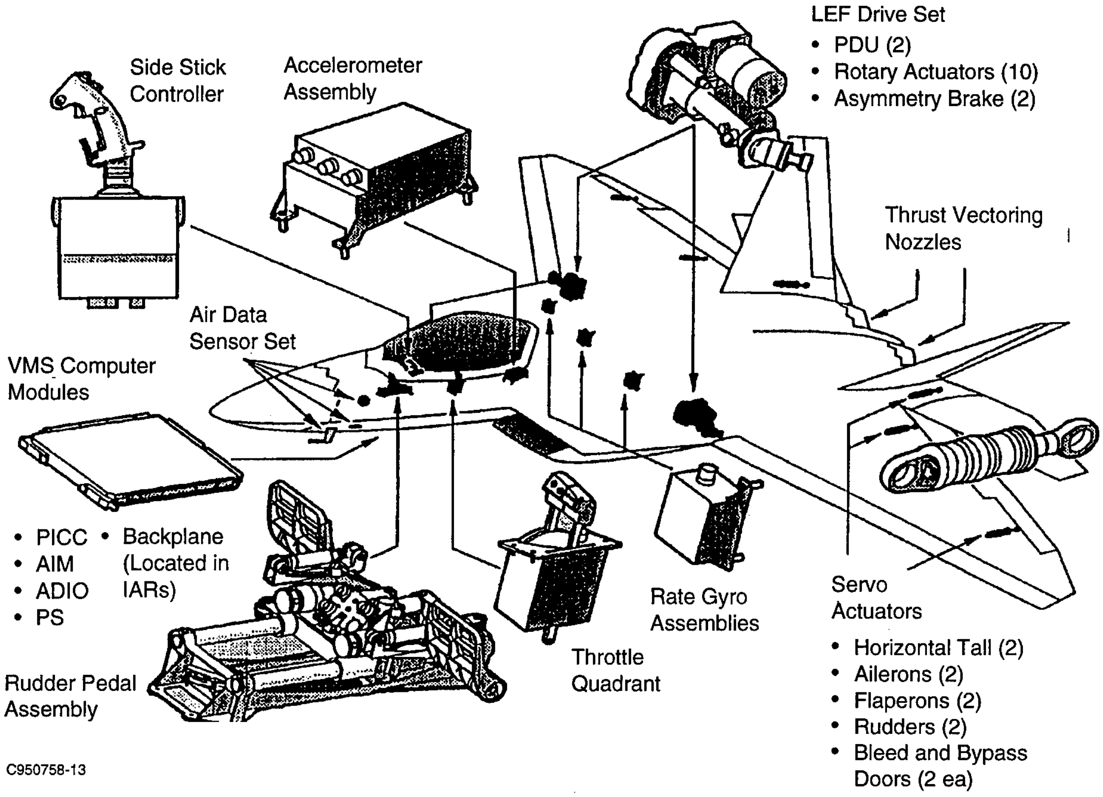


Figure 5-10. VMS Hardware

5.3.3.6 Integrated Functions

Besides providing the basic flight control and engine control functions, there are several integrated control functions provided by the system. They include:

- Integrated flight/propulsion control
  - Thrust vectoring
    - Flight control calculates thrust magnitude and direction
    - Throttle vector commands transmitted to ECU
  - Auto Throttle
  - PSC (future capability)
- Aircraft control at high α using IRS inputs
- Coupling with fuel transfer system to maintain optimum CG
- IVSC coordinates engine starts with APS and engine
- IVSC provides master caution light driver functions for all subsystems in the VMS
- IVSC integrates, controls, and monitors all utilities and subsystems
  - Auxiliary power unit
  - Electrical systems

Table 5-1. F-22 VMS Common Modules

PICC* (Processor, Interface Controller and Communications)	PS** (Power Supply Module)	ADIO *** (Analog and Discrete Input/Output)	AIM *** (Actuator Interface Module)
Manf: Texas Instruments Size: Single-Width Sem-E Weight: 1.3 lb Speed: 2.0 MIPS Clock: 20 MHz Memory: 256K EEPROM 256K SRAM Interfaces: PI-bus, 1553B ICDL, I/O, test connector, system clock, discretes Features: 1750A proc., 12-Bit A/D and D/A, watchdog timer, failure logic and temperature sensor Programmable: Via 1553, ICDL, or PI-bus on A/C, via test connector on test stand	Manf: Boeing ESD Size: Same as PICC Weight: Same as PICC Input: 28 VDC Output: +15 V +5 V +2.2 V Monitoring for over-voltage, undervoltage, and overcurrent Protection for over-voltage and current limiting	Manf: Lear Astronics Size: Same as PICC Weight: Same as PICC 10 AC diff analog inputs 6 DC diff analog inputs 16 Open/GND discrete Inputs Various sensor excitation and accelerometer torque outputs 1 module per FLCS branch	Manf: Lear Astronics Size: Same as PICC Weight: Same as PICC Interfaces: 1 DDV, 2 EHVs Provides total mode control, loop closure, and position sensor Excitation for DDV and EHV actuation Provides eight spare discrete inputs Six modules per FLCS branch

* This module is used by the FLCS, IVSC, and SMS (FDR, EW).
** This module is used by the FLCS and SMS (IVSC).
*** These modules are used only by the FLCS.

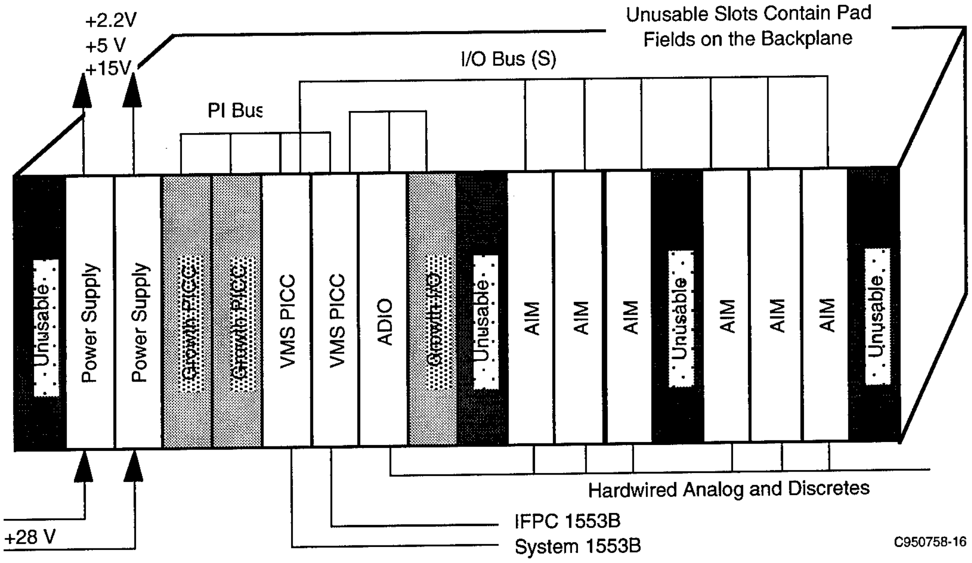


Figure 5-11. F-22 Common Module Rack

- Environmental control system
- Fire detection/extinguishing
- Fuel system
- Hydraulic system
- Landing gear
- Life support system

5.4 RAH-66 COMANCHE HELICOPTER

The Comanche is being developed as the next-generation reconnaissance/attack helicopter. The objectives are very like those of the ATF, increase capability and reduce LCC through the use of an integrated system approach. As was shown in Table 5-1, the Comanche emphasizes several VMS technologies that provide [5.7]:

- Improved flight safety and reliability through the use of an architecture that minimizes failures;
- Enhanced mission effectiveness through selectable control moding to optimize a task, for example, programmed evasive maneuvers to counter threats;
- Increased integration of mission required functions, including flight, engine, navigation, fire control, and cockpit.

5.4.1 RAH-66 Avionics Architecture

The Comanche avionics architecture is shown in Figure 5-12, [5.6]. This system is identified as the Mission Equipment Package. The LONGBOW is a millimeter-wave radar detection and illumination sensor for destruction of antitank missiles. Other sensors include the Electro-Optical Target Acquisition and Designation System (EOTADS) for detection, classification, tracking, and engagement of targets and the Night Adverse Weather Pilotage System, which enhances the pilot's situational awareness to complete night and adverse weather missions. Optical data buses are used to interconnect the sensor system with the mission computer.

The mission computer clusters provide the resources to execute the mission and control the MEP. The interface between the mission computers and displays are also achieved through the use of optical data buses. The cockpit consists of two multifunction displays (MFDs) at each crew station; a right, high resolution, color, active matrix LCD for instrument graphics and map information; and a left monochrome, LCD for textural and sensor video. The cockpit also contains two multipurpose displays at each station. These low resolution, monochrome, LCDs are used for mode selection, radio, status, and emergency backup.

A high-speed data bus couples the mission computers with the ICNIA and INEWS. A MIL-STD-1553 data bus provides the interface between the mission and vehicle systems. The MEP software coordinates overall software activities and performs mission data management and communications functions. The Crew station interface management software provides the interface between the operator-controlled hardware and software and the remainder of the MEP subsystems.

5.4.2 RAH-66 VMS

Although the Comanche does not specifically call any systems a VMS, such functions are readily apparent when examining the system architecture (see Figure 5-13). These functions include the flight control system, engine control system, Air Vehicle Interface and Control Subsystem (AVICS), and portions of the controls and displays.

A functional illustration of the fly-by-wire flight control system is shown in Figure 5-14. The primary FCS performs all flight critical functions. The automatic flight control system handles mission critical functions. It provides Level 1 rating in either VFR or in night/adverse weather flight conditions resulting in high agility or NOE

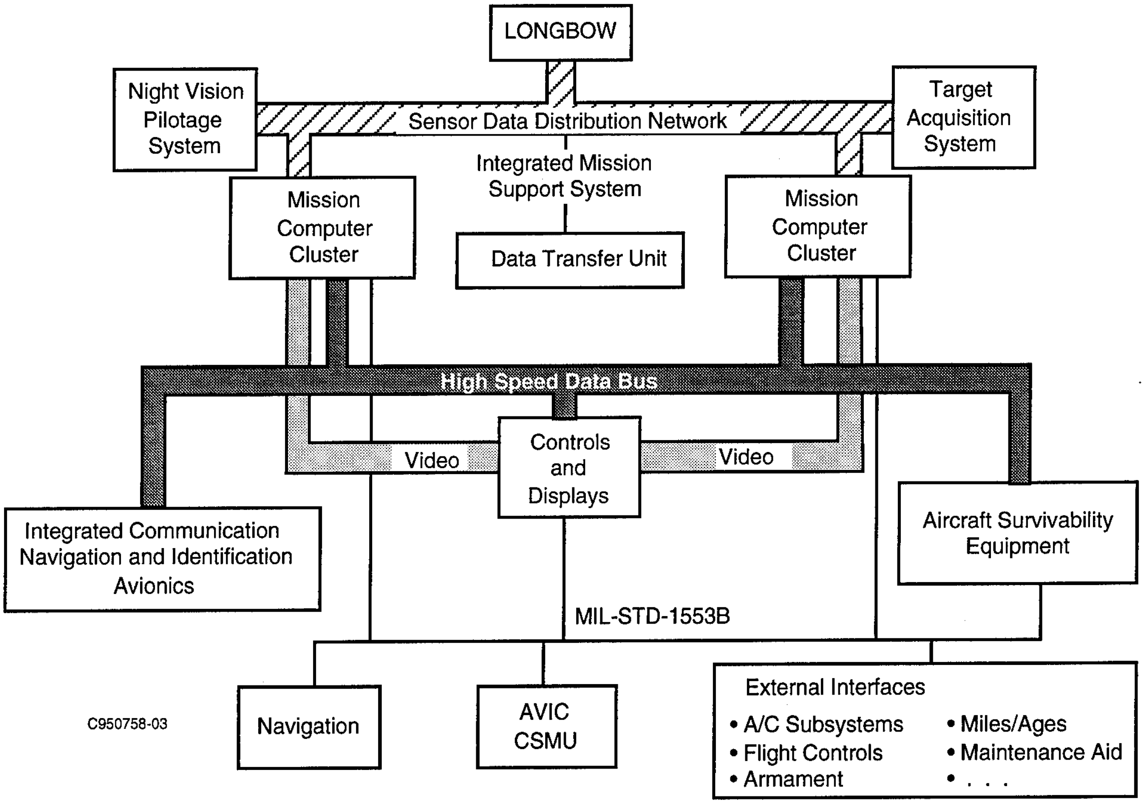


Figure 5-12. RAH-66 Avionics Architecture

and weapon delivery modes. The flight director mission aiding functions include integrated flight and fire control (IFFC), coupled navigation, and envelope cueing functions. The control laws are functionally partitioned so that each of the levels is processed in separate processors.

5.4.2.1 Bus Types

As shown in Figure 5-14, the flight control bus is a triplex 1553B data bus. It interconnects the three flight control computers with the displays, engine control unit, and the three inertial sensors. Two inertial sensors are Flight Control Quality (FCQIS) and the third is Non Quality (NQIS). The system receives its sensor information over the bus. The FC computers are also linked with a MEP 1553B avionics bus to interface with the mission functions.

5.4.2.2 Interfaces

The flight control system is interfaced with several other subsystems. Flight control computer interfaces to the pilot controls and actuator outputs are hardwired. Vehicle state sensor inputs to the control system are through the 1553 data bus. The interface to the AVIC computer is through the avionics data bus. The AVIC provides management of some subsystems. The AVIC units also collect and stores diagnostic information that can be accessed by ground personnel using laptop computers.

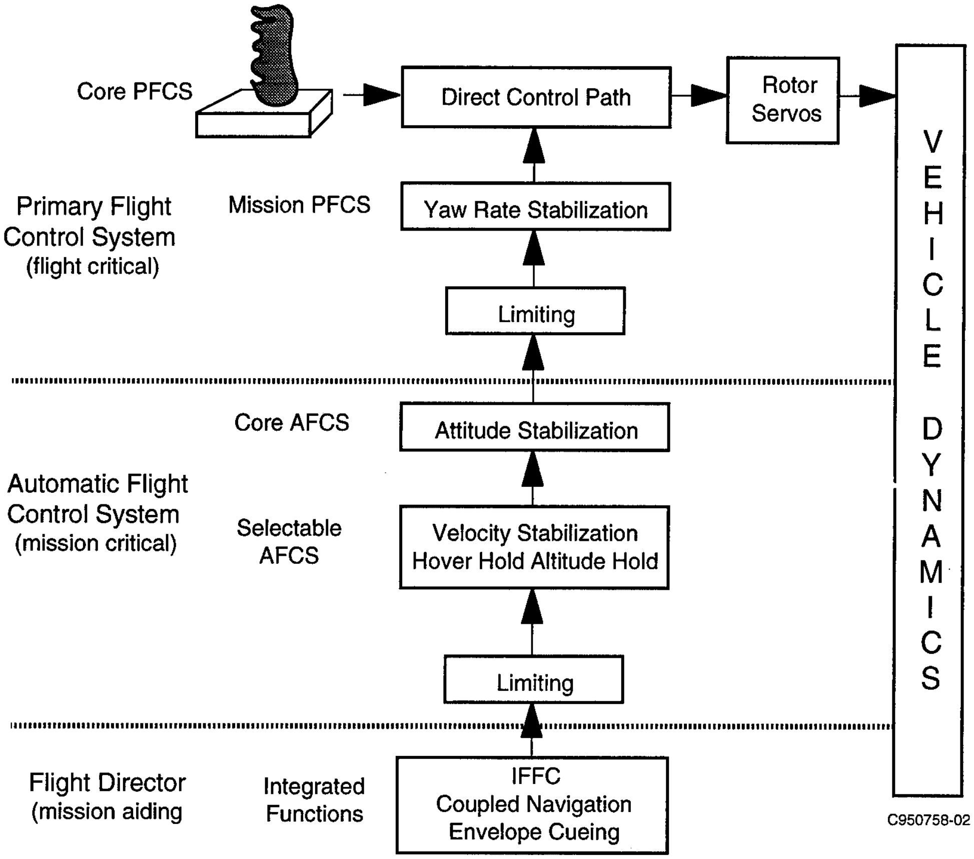

Core PFCS
Direct Control Path
Rotor Servos
VEHICLE DYNAMICS
Primary Flight Control System (flight critical)
Mission PFCS
Yaw Rate Stabilization
Limiting
Core AFCS
Attitude Stabilization
Automatic Flight Control System (mission critical)
Selectable AFCS
Velocity Stabilization Hover Hold Altitude Hold
Limiting
Flight Director (mission aiding
Integrated Functions
IFFC Coupled Navigation Envelope Cueing
C950758-02

Figure 5-13. RAH-66 Flight Control Functional Diagram

The interface with the crew station subsystem provides pilot and copilot selections and inputs through the displays and switches. The flight control system also provides information to the MEP to support NOE, IMC, and cruise flight modes. This data consists of airspeed, inertial data, engine data, control inputs, and weight-on-wheels.

The flight control system also provides inertial sensor data to the MEP navigation subsystem. This information includes data derived from magnetic flux valves. This data is available from all three FCS channels, with two channels having medium grade performance and the other channel having high grade performance. An independent Kalman filter is calculated using GPS, doppler, and INS for each FCS inertial sensor.

5.4.2.2.1 Air Vehicle Interface and Control System—The Air Vehicle Interface and Control System (AVICS) system consists of dual control units that provide interfaces to the aircraft vehicle subsystems. The units are interfaced to the mission computer clusters through the MIL-STD-1553 bus. Each AVICS is coupled to the three aircraft power controllers through three RS-422 links. AVICS Unit 1 couples the crash survivable memory unit (CSMU) and portable download unit (PDU) through RS-422 links. This unit is also directly coupled to the ECS mechanical system, provides excitation to the aircraft transducers, receives aircraft transducer signals and is supplied left E-Bay power supply status information. AVICS Unit 2 is coupled to the rotor tracking system through an RS-422 data link, is also similarly coupled to the aircraft transducers, and receives right E-Bay power supply status information.

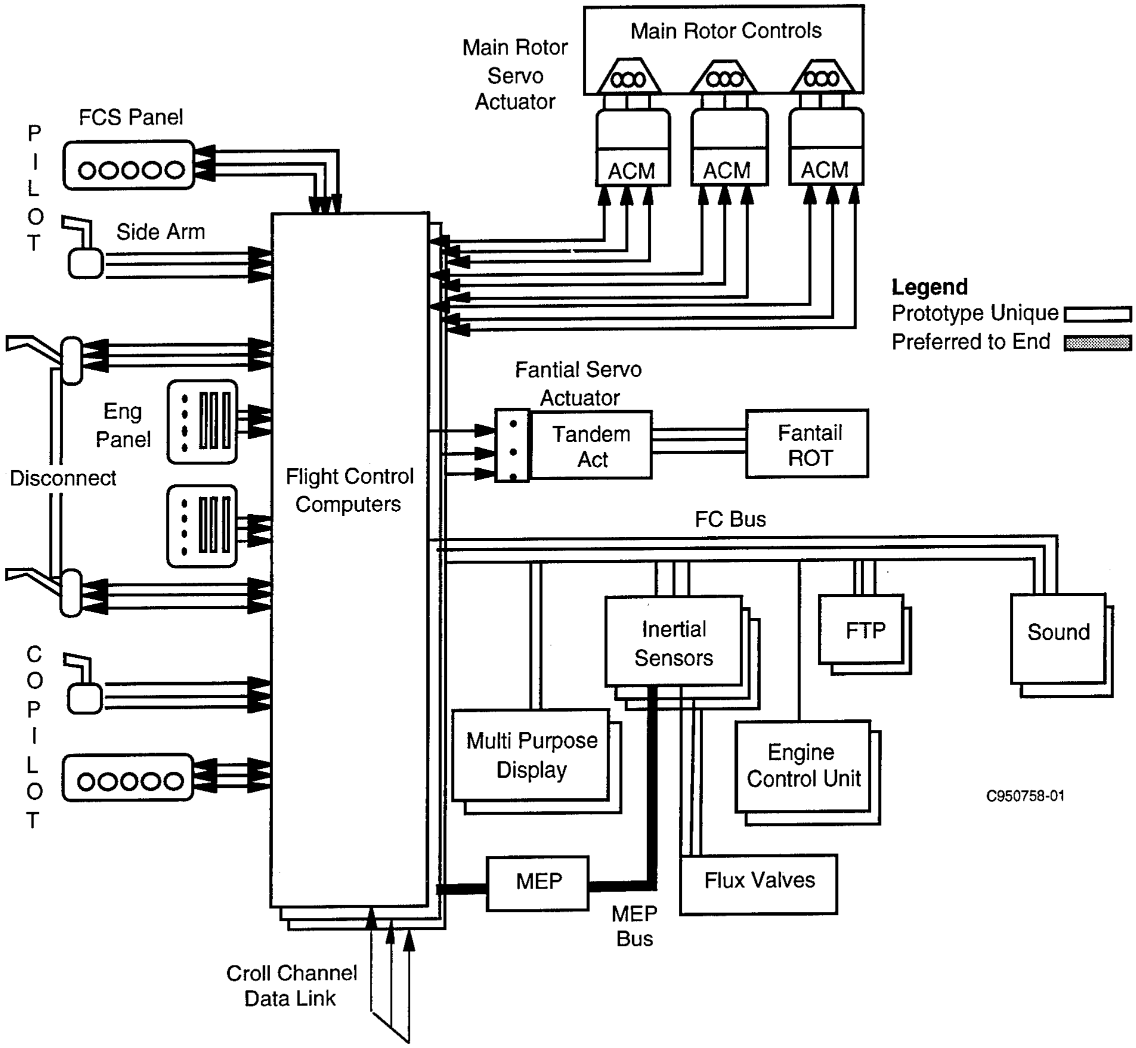


Figure 5-14. RAH-66 Flight Control System

5.4.2.3 Sensor/Actuator Types

All actuators are jam-resistant, dual hydraulic actuators with redundant control valves. The side arms controller provides pitch, roll, and yaw control with limited control authority in the vertical axis. Vehicle state sensors consist of omnidirectional very low speed air data sensors, and inertial sensors aided by doppler radar altimeter, flux valve, weight-on-wheels, radar, and GPS.

5.4.2.4 Redundancy

The flight control system is a triply redundant system to provide fail-op/fail-op capability. The system has full flight-critical fault detection and self-reconfiguration capability. Each FCC uses a self-checking pair of MIL-STD-1750A processors. The engine control system is a full-authority dual digital system. In the event of the loss of both channels, backup control would be performed by the FC computer. The FCC BIT provides diagnostic functions that isolate faults down to the engine module. Electrical and hydraulic power sources are dual redundant to provide the required levels of reliability. Critical components are physically separated from each other to enhance ballistic protection. The system is also shielded to provide hardening against EMI, lightning, nuclear radiation, and high power microwaves.

5.4.2.5 Avionics Modules

The common modules utilized in the Comanche follow the principles utilized in the F-22 but are not as extensive. Most modules are conductively-cooled SEM-Es. Racks are cooled with chilled air. The modules are removable at the unit maintenance level. The FCC modules are non SEM-E e for reduced cost.

5.4.2.6 Integrated Functions

The IFPC provides coupling of the engine control system to the flight control system to provide anticipation of rotor load demands and to control rotor speed under varying load conditions. This system when coupled to the weapons delivery system to provide an integrated flight fire control (IFFC) function produces a more agile and responsive aircraft and also reduces flight path excursions. This, in turn, produces more accurate weapon pointing, tightening shot patterns and greatly reducing aiming errors. Pilot workload is also significantly reduced by this integration. The Integrated Flight/Navigation capability provides a three-dimensional flight director. The Envelope Cueing capability warns the pilot to avoid exceeding the service flight envelope using voice and helmet displays.

5.5 EXPERIMENTAL AIRCRAFT PROGRAM (EAP)

The EAP program was a UK effort to demonstrate specific technologies required for the European Fighter Aircraft (EFA). The program was a single aircraft program intended to flight demonstrate a variety of critical technologies, including structures, advanced aerodynamics, advanced control, systems integration, and multi-function displays. The aircraft first flew in August 1986 and was the first flight demonstration of an integrated VMS-type system. The impetus for the program was airframe size and the emerging digital control technologies.

5.5.1 EAP Architecture

The basic architecture of the EAP aircraft is shown in Figure 5-15. The architecture was composed of three major systems:

- Limited avionics suite, including communications, navigation, and display management;
- VMS, including flight control, propulsion control, and utilities management system (UMS);
- Reversionary display system.

The avionics bus was a MIL-STD-1553B bus.

5.5.2. EAP VMS

The EAP VMS is shown in Figure 5-16. The system consists of a digital flight control system interfaced to the UMS. The primary control areas are coupled by data buses and conventional wiring; however, there is no fusing of major control functions. Thus, the system is an example of a physically but not functionally integrated VMS.

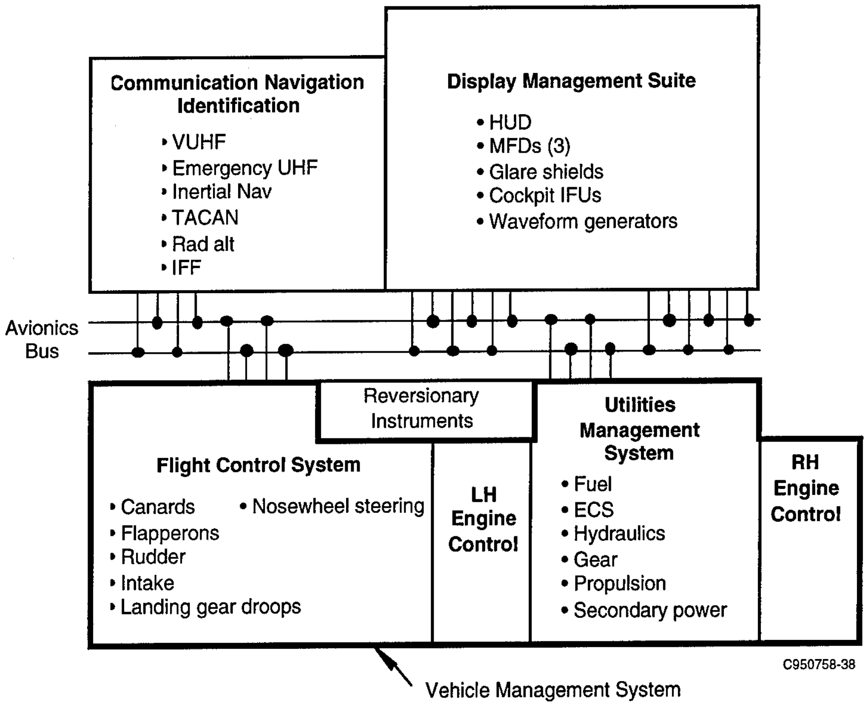


Figure 5-15. EAP Architecture

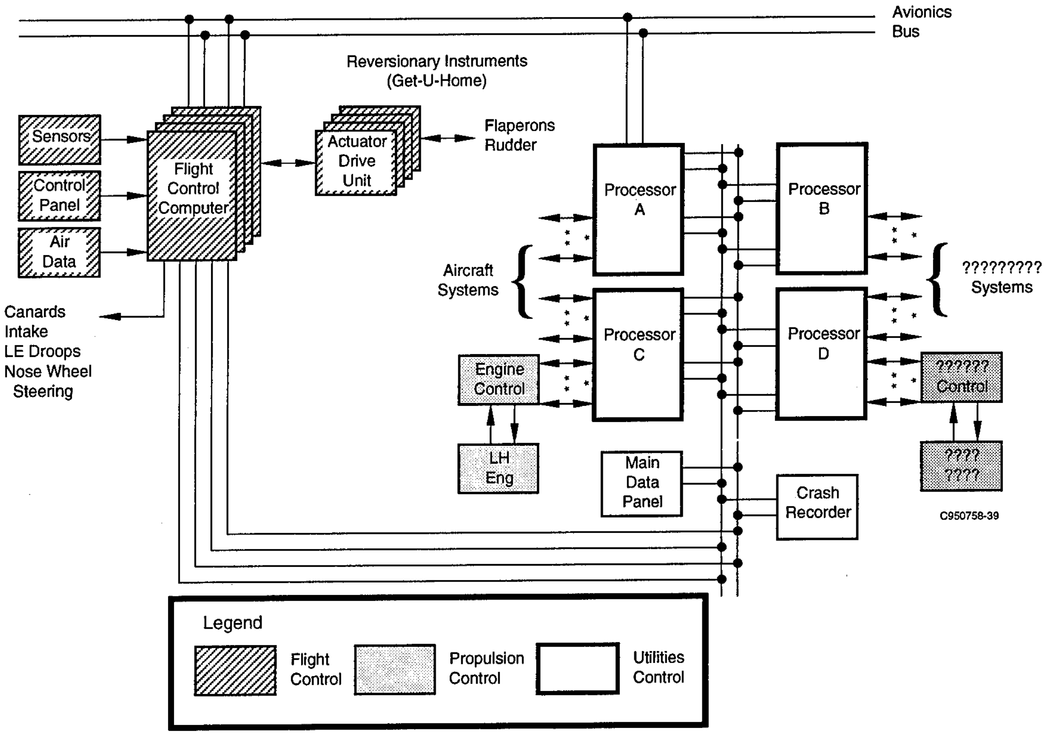


Figure 5-16. EAP VMS

The requirements for the VMS were to:

- Reduce aircraft wiring and provide improved interface capabilities,
- Provide increased automation to reduce pilot workload,
- Reduce the number of hardware items to minimize weight and volume,
- Provide a means of controlling and monitoring the suite of utility systems.

The four processors in the UMS system provide control and management of the aircraft utility systems. Processors C and D provided fuel gauging/level sensing, fuel transfer, engine fuel feed, and refuel, defuel, and dump functions. Processor B provides ECS/cabin temperature control. Secondary power and hydraulics control was provided by processors C and D, and all four processors provided undercarriage monitoring, weight-on-wheels, normal, and emergency brake control.

5.5.2.1 Bus Types

As shown in Figures 5-15 and 5-16, the entire system is mechanized with a MIL-STD-1553B bus system. Control of the avionics bus is provided by "waveform generators" connected to the avionics bus. Processors A and B perform the bus control bus function for the UMS.

5.5.2.2 Interfaces

The FC computers are coupled with the utilities bus and the avionics bus. Air data and flight control display information is exported to the multifunction displays. Air data is also exported to the utilities system via the 1553B interface. The FC computers also interfaced to the reversionary instruments for get home capability. Sensor inputs and commands to actuators are hardwired. The UMS is coupled to the avionics system through processors A and B. This interface provides data for the multifunction displays and receives pilot commands for the UMS system. The UMS bus also provides get home information to the reversionary displays in case of an emergency. The reversionary information is provided by a direct link from processors C and D. Maintenance Data Panel is also connected to the UMS bus. This system permits the monitoring of the system's health and provides a means for logging maintenance data.

5.5.2.3 Sensor/Actuator Types

Aircraft motion was sensed by four strap-down inertial units that provided three axis body rates and accelerations as well as three axis attitude and inertial information. Air data from the pitot/static system was provided by two dual air computers. Four Airstream Direction Detectors (ADDs) fed incidence and slideslip inputs into each of the four flight control computers.

Actuator commands for the flaperons and rudder were fed to four actuator drive units which in turn drive the respective actuators. The actuators had a tamden hydraulic drive controlled by an electrohydraulic first stage. Half of the quad actuator was driven by one hydraulic system and the other half by the second hydraulic system. The canard, leading edge droop, intake, and nose wheel steering actuators were driven directly from the computers.

5.5.2.4 Redundancy

The FCS was a quad digital system that was interfaced to the dual-channel avionics system and the dual UMS. The digital engine controllers were dual redundant. No mechanical reversionary backup system was utlized. Since the fault tolerance requirements of the UMS were less stringent than those of the flight controls, it was mechanized in a dual redundant configuration.

5.5.2.5 Avionics Modules

The system was implemented using a modular approach. The basic building block is a _ ATR module, shown in Figure 5-17. The modules, in turn, are mounted within an LRU. Fifteen module types satisfied the requirements for the seventy-two cards utilized within the system. Modules consisted of processors, power supplies, memory, 1553B interface, and a variety of I/O cards. The types of I/O cards were analog, frequency, and discretes. Over 600 signals were identified in the VMS system. The processor cards were implemented with Z8001 processors. The processors were programmed in PASCAL.

5.5.2.6 Integrated Functions

Although the system was physically integrated, there were no integrated control functions implemented within the EAP system. At the time the system was developed, the reasons for not mechanizing any integrated control modes was the differences in integrity among the various subsystems. However, the capability to incorporate integrated functions was available.

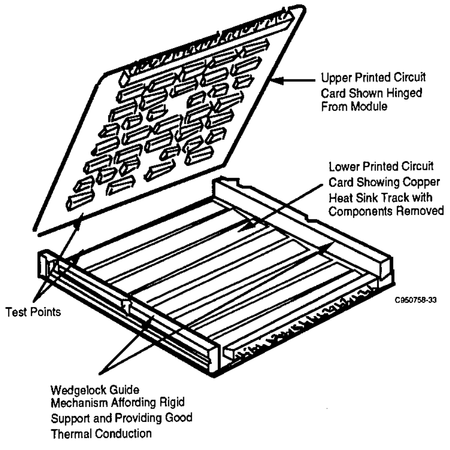

Upper Printed Circuit Card Shown Hinged From Module
Lower Printed Circuit Card Showing Copper Heat Sink Track with Components Removed
C950758-33
Test Points
Wedgelock Guide Mechanism Affording Rigid Support and Providing Good Thermal Conduction

Figure 5-17. EAP Standard Module

5.6 REFERENCES

[5.1] McWha, J., "777 Systems Overview," Seattle, WA, 14 October 1994, WG-12 Working Group Meeting.

[5.2] Driscoll, K., and Hoyme, K., "SAFEbus™," 11th Digital Avionics Systems Conference, 5-9 October 1992.

[5.3] Leonard, B., "YF-22/F-22 Vehicle Management System," Seattle, WA, 14 October 1993, WG-12 Working Group Meeting

[5.4] White, J. E., "Advanced Avionics Architecture—Support Challenges for the 21st Century," Anaheim, CA., AUTOTESTCON, 24-26 September, 1991.

[5.5] Warwick, G., "Clever Cockpits," Flight International, 8-14 January 1982, pp. 19-24.

[5.6] Rosen, Dr. K. M., "An Overview of Technologies Embodied in the Comanche Program,"Thirty-Fifth Israel Annual Conference on Aerospace Sciences," Tel Aviv, Israel, 15-16 February 1995.

[5.7] Landis, K. H., Davis, J. M., Dabundu, C., and Keller, J. F., "Advanced Flight Control Technology Achievements at Boeing Helicopter," *Int. J. Control*, Vol. 59, No 1, 1994.

[5.8] Colucci, F., "Stealthy Scout—The RAH-66 Comanche," *Air International*, July 1994, pp. 38-44.

[5.9] Moir, I., and Seabridge, A. G., "Management of Utility Systems in the Experimental Aircraft Programme," *Aerospace*, September 1986, pp. 28-34.

[5.10] Moir, I., "Application of Digital Technology to Aircraft Utility Systems," 8th Digital Avionics Systems Conference, San Jose, CA, 17-20 October 1988.

[5.11] Middleton, D. H., "Avionic Systems," Longman Scientific & Technical.

CHAPTER 6

VEHICLE MANAGEMENT SYSTEM ENABLING TECHNOLOGIES

6.1 INTRODUCTION

The composition of a VMS is discussed in Chapter 2. The purpose of this chapter is to define various technologies that can be applied to an IVMS to allow enhanced capabilities with improved reliability and lower cost.

In many instances, the technology has its origin in R&D conducted for integrated avionics systems: standard digital interfaces, microprocessors suitable for use at the subsystem level, and structured software methods. The ingredients that make these electronic and computer science technologies applicable to IVMS development are advancements in reliability and fault tolerance. In addition, the design tools that allow these enabling technologies to be applied are readily available as design methodologies, specifications, and software packages.

6.2 DIGITIZATION OF INTERFACES

6.2.1 Distributed Structural Sensors

The integration of active controls, structures, and aerodynamics is leading to the development of new or improved integrated system concepts that can affect vehicle weight and performance. Among these concepts are control of reduced torsional stiffness wings, relaxed flutter margins, active load alleviation, relaxed structural design margins, vibration suppression, and real-time damage reconfiguration. There is also growing interest in the application of smart structures to perform health monitoring, real-time damage assessment, and load history. Key to these advancements is the development of networks of very small, lightweight sensors distributed over the structure. These sensors might be strain gauges or accelerometers fixed to the skin or structure or sensors embedded within the material, such as fiber optics in a composite material. However, the effect of embedded fibers on composites is still not completely known, and the interconnection and repair problems require more research. Other sensor types that hold potential are acoustic, ultrasonic, and piezoelectric. Figure 6-1 shows a schematic of how these types of sensors might be mechanized in the wing of an aircraft. The network of sensors can be utilized to determine the structural shape, identify structural modes, measure loading effects, and identify flexible mode frequencies and mode shapes. The distributed sensors can also identify a damaged structural area, which, in turn, can invoke a control reconfiguration mode to enable continuation of safe flight.

A challenge for the IVMS is the processing of the vast amounts of information received from the sensor network.

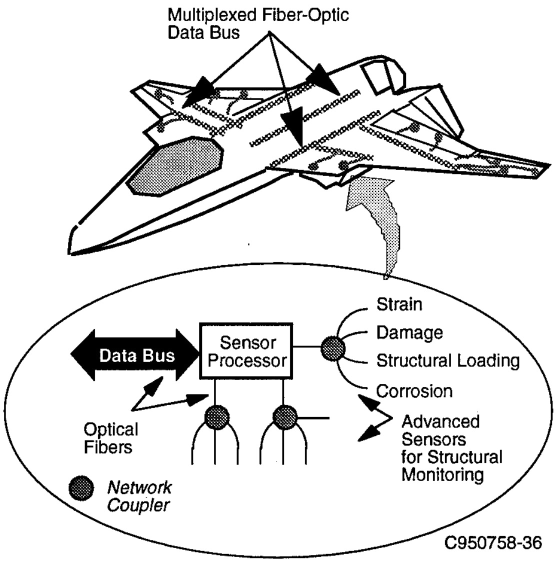


Figure 6-1. Distributed Structural Sensors

6.2.2 Digital Sensors

The future use of common digital interfaces will result in significant savings in sensor electronics cost, size, and power. The ability to make these interfaces simpler than analog, which requires A/D conversion electronics, also enhances the reliability of future systems.

For sensors with inherent digital outputs, the results are easy to achieve. For analog sensor outputs, the conversion to digital must be done effectively to achieve the goals of low cost and high reliability.

One example of the benefits of a common digital sensor interface is the work done on the ring laser gyro. As shown in Figure 6-2, a common digital interface will allow the user to eliminate computer cards devoted to conversion electronics. Another benefit is the increased protection from high energy interference, or HIRF. This can be achieved by simply containing the lengthy runs of conduction material, such as electric cable, to short communication runs in shielded environments.

Once the capability to produce digital output sensors and conversion of analog devices is established, new interface standards can be established. These standards will create improvements in life-cycle costs due to plug compatibility of sensors. This will take advantage of changes in technology and overcome product obsolescence.

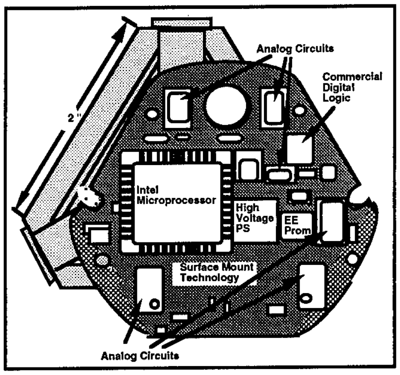


Figure 6-2. Digital RLG Electronics

6.2.3 Vehicle Configuration

The vehicle configuration state as defined herein consists of the vehicle systems that alter the external configuration of the vehicle. They consist of such systems as doors, landing gear, wing sweep, and vectoring nozzles. Except for the vectoring nozzles, these systems all operate open loop. The vectoring nozzle is hydraulically driven through actuators and utilizes position sensors in the control loop. The control function for the vectoring nozzle is contained in the engine controller or can be embedded in the IFFC function. The requirements for the vectoring nozzles are equivalent to those for the control surfaces as they are providing an active control function.

External doors are hydraulically actuated and are manually controlled. Proximity or microswitches are utilized to indicate open or closed positions. Landing gear are likewise hydraulically actuated and are also manually controlled. Two position switches are used to indicate gear up or down position. A weight-on-wheels sensor is also utilized in military aircraft to indicate a landed condition that might, in turn, allow thrust reversers to be actuated. For swing wing applications, a hydraulic motor driving a screw jack is typically employed. A position switch is utilized to indicate position of the wing.

The moreelectric concept is examining the replacement of hydraulic actuators with electrically driven effector devices. The actuation devices discussed above are being considered in these investigations. Issues are whether the necessary rates, torques, and responses can be provided at equivalent costs. The more electric approach will impact how the utility subsystems are integrated in terms of power generation and distribution to the actuators.

6.2.4 Actuator Systems

Actuator systems for FBW applications require high response rates and sufficient resolution to meet performance requirements. These requirements translate into actuator bandwidth and word length. Because of these stringent demands, all systems up to the present have utilized analog loop closure of the actuator.

There is a current trend toward the use of pulse-width modulation to reduce system cost. Two position valves are used to control the actuator rather than the popular proportional valves. The pulse repetition rate is set up to coincide with the desired response capability, and the pulse width controls the flow to the actuator (varies as a function of error).

The control electronics utilizes an oscillator generating a triangular waveform signal. This signal is summed with the analog error generated by the difference between the commanded and feedback voltages. The resulting signal is fed to comparators that generate the PWM output.

Full digital loop closure has been utilized on an experimental basis, but application has been hampered by resolution and update rate problems. Other recurrent limiting factors are the availability of a digital position transducer and complexity of the support electronics.

There is a trend toward moving the actuator electronics to a location near the actuator, at least for subsonic applications where ambient temperatures do not become extreme. This smart actuator concept has the effect of reducing wiring complexity and weight because loop closure, cross strapping for redundancy, and failure monitoring can be accomplished locally. Command and actuator status information can be transferred over a data bus.

The various actuator systems and actuator type options will impact the way the IVMS architecture is mechanized. The important considerations are fault tolerance and safety issues.

The trend in actuators is toward direct drive (DD), electrohydrostatic (EHS), and electric. The DD actuator, as described above, uses electric actuation to drive the main power directly. The EHS actuator uses signals and electric power to control and power a self-contained hydraulic power unit. In the electric actuator, hydraulics is eliminated and replaced with electric actuation devices, as described in Subsection 6.2.3.

6.3 PROCESSORS

Digital processors needed for future IVMSs are characterized by modest computational throughput, low latency and jitter, efficient multitasking, and hard-real-time response. The processors must also be highly reliable, readily available, and testable in ways not typical of conventional processor applications. In addition, the processor must be affordable to procure and maintain.

The availability of a wide range of processors capable of implementing many of the critical VMS functions has been made possible by the large investment made by the defense and commercial sectors during the past decade. The specific characteristics of a processor that lends itself to application to a VMS are discussed below.

The processor must be deterministic and synchronous to ensure that the same input will unequivocally yield the same output in a redundant configuration. This synchronism requirement means that there cannot be boundaries between circuitry that use different clocks within the processor and all signals external to the processor must be synchronizable to the clock that the processor will use to sample the signal. Typically these input signals are interrupts and reset. Reset may be asynchronous when it goes active but may release in synchrony with the processors' clock(s). The synchronism requirement on reset can be relaxed if a redundant configuration does not do cycle-by-cycle comparison of the redundant processors' actions. The same cannot be said for other inputs. Even loosely coupled, inexactly voted redundant processors cannot use asynchronous input signals because it is not feasible to verify the correct operation of such a configuration.

This consistency of operation must also be exhibited by the processor during test and integration. Ideally, any controllability/observability mechanisms employed during test and evaluation would be noninvasive; that is, these mechanisms would not affect the normal processor operation. Although totally noninvasive instrumentation may not be currently feasible, minimally invasive techniques are the goal. This is particularly difficult with the emerging class of processors that use large, on-chip cache memories and/or high-speed memory buses. The large on-chip caches prevent the passive observation of program execution because much of the operation occurs completely within the processor chip. These new processor chips will have to add internal hardware support to replace this lost observability. Similarly, high speed memory buses can be affected by the capacitance added when a "passive" probe is attached to the bus. Again, the inability to noninvasively observe the memory bus has to be compensated for by new support hardware within the processor.

Embedded processors such as those used in VMS systems have their programs stored in nonvolatile memory. Nonvolatile memory is used so that programs are not lost when power is interrupted. This nonvolatile memory interferes with the breakpoint and single mechanisms used by most software debugging tools. This is another area that requires on-chip hardware support.

The software and hardware environment accompanying the processor must be extensive to ensure that the test and integration phase of the VMS development are not brought to a standstill by undocumented behavior. In addition, all test and debug hooks and modes must be positioned and locked out of normal flight operation to ensure integrity.

Reduced instruction set architectures are well suited for VMS applications because they are relatively easy to verify and they keep all hardware elements continually exercised. This greatly reduces the risk of long latency times for detecting faults in seldom used hardware and microcode.

The memory management unit of processors employed in a VMS typically should not use virtual memory features, as the memory size requirements are modest and virtual memory implementations are difficult to verify. However, the memory management unit may be needed to enforce memory protection.

Similarly, the interrupt structure must be kept as simple and as fast as possible. The jitter in interrupt latency is typically a more difficult problem than the magnitude of the average latency. Multiple interrupts are very difficult to verify, and nested interrupts are practically impossible to verify. These problems suggest that only two interrupt sources should be allowed: a clock tick and immanent-power-fail. The clock interrupt paces the processor and is the only interrupt used for normal operation. The power-fail interrupt should never return; the processor only resumes after the subsequent power-on reset.

The trend toward lower voltage processors facilitates higher speed within current operating temperature limits but exposes the processor to single-event upsets (SEUs) due to EMI or atomic level particle-induced SEU. To prevent power dissipation from exceeding the thermal limits imposed by the VMS environment while increasing processor throughput requires that each processor operation (signal or bit change) use less energy. Lowering the energy used for normal operation places it closer to the energies of external sources. This means that more of these external sources of noise energy can cause the processor to malfunction transiently. Designers of future systems will have to take this increased transient failure rate into account in their system reliability calculations and designs.

Ideally, the processor chosen for a VMS would be in a family of processors to reduce the nonrecurring engineering needed to establish the design, development, test, and integration environment when a processor upgrade is needed. The family approach provides a stable and long production run necessary for low recurring costs and assured availability for the system life cycle and beyond. Processor obsolescence is a particularly difficult problem for current commercial processors because the product lives have been averaging two years or less. This is unacceptable for future military electronic systems that will have life cycles of 20 years or more.

6.4 INTERCONNECTS

6.4.1 General

An IVMS is composed of sensor interfaces, computing devices, and actuator interfaces, and the glue that holds them together is the interconnecting network structure. This structure can be, and usually is, an ad hoc heterogeneous collection of analog and digital networks of various topologies and protocols. Since the first VMS designs, there has been a continuing decline in the use of analog links, which are being replaced by digital networks. Digital signals are more reliable and dependable, require less maintenance attention, can be multiplexed to reduce wiring, and consume less power than analog signals (for the 0.05% accuracies typically used by VMSs). Analog links are now relegated to the periphery of the VMS to connect to analog sensors and actuators. Smart integrated sensors are shrinking the length of this last analog link to less than a millimeter. This transition from analog to digital is an enabling technology for IVMS because analog links are difficult to share dependably.

6.4.2 Characteristics and Requirements

The networks used for VMSs have some unique characteristics and requirements. VMSs are real-time embedded control systems that typically can contain high-speed (e.g., 80-Hz) loops and need to have very high dependability. Compared to more general-purpose digital networks, VMS messages are very small. These characteristics engender the following requirements.

6.4.2.1 Low Latency and Jitter

A major concern of control system designers is the phase lag introduced by the control system. Latency (the time span between when a message is ready to transfer and when the message arrives at all intended destinations) and jitter (the variance in latency) are major contributors to the total lag. A large jitter is more troublesome than a large constant latency because compensation for it is much more difficult or impossible. In general, jitter is caused by asynchronism. This is one reason why networks using asynchronous arbitration should be avoided for VMSs. An even larger source of jitter is the full-frame jitter that occurs if a processor is running with a period similar to the period of the network or other processors but is not synchronized with them. This jitter is eliminated if all the processors and network(s) are synchronized together.

6.4.2.2 Efficiency

There are two places where efficiency is of concern: media and processing. The high repetition rate of small messages means that any message overhead consumes a high percentage of the available resources. To conserve media bandwidth, VMS networks should have little or no bits of overhead per message and should use an efficient encoding scheme (e.g., Manchester has half the efficiency of NRZ). Table 6-1 shows the media efficiency for a mature standard network (1553), two emerging standards (the SAE bus and ring), and for a research network (RealNet) for messages consisting of 2 bytes and 32 bytes.

Table 6-1. Network Efficiencies

	2 Bytes	32 Bytes
MIL-STD 1553	10%	36%
SAE AS 2 Bus	7%	36%
SAE AS 2 Ring	3%	22%
RealNet	55%	94%

To conserve processing resources, the network should be able to handle message transfers with minimum intervention of a host processor and without needing a dedicated protocol processor. A network that is capable of performing (some) operations independently of a host processor is sometimes called "autonomous."

One source of needless processing inefficiency and latency is implementing a too-complex protocol stack. The SAE has recognized that the seven-layer OSI protocol stack model used by many commercial (office) LANs is not appropriate for real-time applications. In its place, the SAE developed a four-layer model. The OSI stack was developed for an environment where a very large number of devices can be connected in a huge number of different configurations. The configurations are also fairly dynamic. A large OSI-style network might have daily configuration changes. In comparison, VMS networks have a tiny number of different devices and a small number of pre-defined configurations. The definitions of new configurations are strictly controlled and are nearly static compared to OSI networks.

Excess complexity in a protocol stack is not only a source of inefficiency and latency, but the complexity is also a detriment to ensuring dependability. Therefore, networks that require a complex protocol stack, such as those designed to be compliant with the OSI model, should be avoided for VMS use.

At the other end of the spectrum, there are now networks that can offload some of the functions that have been traditionally done by processors and/or other dedicated hardware. These include:

- Freshness,
- Inter/intrachannel synchronization,
- Processor fault masking support,
- Task scheduling control.

A common task that needs to be performed by real-time control systems is determining the freshness of inputs used for computation. Erroneous output may result if inputs are used that are too old; or even worse, old inputs are mistaken for fresh inputs. Freshness determination traditionally has been a time-consuming software task. Networks can support freshness determination by not transferring data that is not fresh and/or by autonomously time tagging the data.

Other common VMS tasks include intrachannel and interchannel synchronization. Intrachannel synchronization is used to minimize latency and jitter in the path from the sensors to the actuators. It is also used to minimize skew and divergence to allow cross-channel voting and equalization. This typically has been done by a combination of software and dedicated synchronization hardware. Networks can provide this synchronization. Thus, these networks can eliminate the need for dedicated synchronization hardware and can offload much of the synchronization load from the processors.

The dependability requirements of most VMS subsystems necessitate some form of fault tolerance. The reconfiguration in response to faults typically has been the responsibility of software. (Hardware-only mechanisms typically have been used only for masking.) There are now network protocols that can autonomously substitute one processor's messages for another faulty processor's messages and/or can disconnect a faulty processor.

Networks can even take responsibility for processor task scheduling control. This mechanism synchronizes processor tasks to the network and (by transitivity) synchronizes between tasks running on different processors. This is done by having network hardware replace the traditional real-time clock tick interrupt with an interrupt that is synchronous with the rest of the network's operations. The network's hardware can even supply an ID for each interrupt occurrence to indicate which tasks should be run.

6.4.2.3 Isochronism and Synchronism

Most VMS control systems are designed to execute in a small number of fixed period loops. This isochronism is characteristic of sampled data control systems. Therefore, a VMS network should support isochronous operations. Only recently have networks been created that autonomously support isochronous operations; that is, they can guarantee the repetition period of their operations to within tight jitter tolerances. By synchronizing processors to the network, the processors can inherit this isochronism.

Networks currently in wide use do not have support for the multiple synchronous channels used by most fault-tolerant VMS subsystems. The networks do not support, within themselves, the synchronization or the cross-channel data exchange needed for these multiple-channel subsystems. This means designers typically are forced to assemble an ad hoc heterogeneous system consisting of one network for intrachannel communication, another network for cross-channel data links, and yet another mechanism for channel synchronization. Network designs are now in the research stage that can combine all these functions into a single mechanism that is more efficient and easier to verify.

6.4.2.4 Dependability

Most VMS subsystems have some sort of dependability requirements. Some of these requirements are very stringent. The various aspects of dependability to consider are defined in Subsection 2.11.3. The original generation of VMS networks only addressed faults that caused individual bit errors within messages or omission faults. These errors were typically detected by parity and timeouts, respectively. Newer networks have increased error coverage with the addition of CRCs or the addition of multiple paths with comparators or voters. Some newer networks also have recognized that message data is not the only thing that can be faulty. They have added fault tolerance mechanisms to cover faults in control signals, addressing mechanisms, protocol engines, etc.

6.4.2.5 Determinism

An important design consideration for systems with stringent dependability requirements is the method(s) used to ensure that the design meets these requirements. Determinism is an essential characteristic of a design in order to have some degree of confidence in its dependability. Determinism is the characteristic of a system that allows an accurate prediction of its future behavior given knowledge of its current state and the sequence of its inputs (including unintended/erroneous inputs). Without determinism, nothing meaningful can be said about a system's future behavior; and therefore it cannot be said to be dependable. The major source of nondeterminism in VMSs is asynchronism. Asynchronism at the lowest levels of the design can lead to metastability errors. Asynchronism at higher levels causes the permutations of event orderings to explode to the point where the system's behavior cannot be analyzed. A VMS network must minimize the number of asynchronous interfaces and the magnitude of the effects caused by asynchronism.

6.4.3 Types of Networks

The major characteristics of a network can be divided into two aspects: topology and protocol.

A network's topology is the structure or general layout of its media and how nodes (devices) are connected to it. The types of topology are:

- Link = just two nodes, directly connected to each other;
- Bus = one length of medium to which all nodes are connected with stubs;
- Ring = nodes are connected in a circle, each node linked to two neighbors;
- Hub/star = all nodes are connected to a central point;
- Mesh = each node is directly linked to more than two other nodes.

Overwhelmingly, the most common topology is the bus—so common in fact, that some people erroneously call all topologies a bus.

A protocol is the set of rules for using the media and other resources of the network. Most often, network protocols are classified by their media access control (MAC) sublayer of their protocol stack. Some of the most popular MAC protocols include Carrier Sense Multiple Access (CSMA), Time Division Multiple Access (TDMA), command/response, and token passing.

Networks can also can be divide into intrabox and interbox types. Intrabox networks, as the name implies, stay within one box (LRU, cabinet, etc.) and thus are usually about a meter or less in length. Interbox networks are typically more than a meter in length and connect multiple boxes together.

6.4.3 Inter-Box

6.4.3.1 MIL-STD 1553 and 1773

The most common interbox network for military aircraft is the MIL-STD 1553 bus. This is a command/response bus with a central controller. Messages can be up to 32 words of 16 bits each. Its raw 1-mbit/s data rate can support about 5 to 700 Kbit/s of real throughput. The dependability weakness of the central controller(*) and the limited throughput have made this bus obsolete. A popular 1553 derivative is the STANAG 3910, which has ancillary data transfer with a raw data rate of 20 mbit/s. The 1773 is a fiber-optic version of the 1553. A new version of 1773 will match the data rate of 3910.

(* While the 1553 standard defines a way to have multiple controllers. The U.S. Air Force has precluded the use of this standard feature. Implementers of 1553 typically still have a standby controller and use a nonstandard method to switch between the controllers. One research project created a bus that allowed many nodes to share the controller job on a per-message basis without violating any of the 1553 standard, including the Air Force restriction.)

6.4.3.2 ARINC 429 (Digital Information Transfer System, Mark 33)

In popularity and age, the commercial avionics equivalent of the 1553 is the ARINC 429 [6.7]. The 429 has a single transmitter and multiple receivers. Data transfers occur by means of 32-bit transmissions. Eight bits of each transmission is a "label" that identifies the contents of the remainder of the word. The data itself may consume as many as 23 bits of the 32-bit transmission. Generalized formats for binary, discrete, and character data have been defined. A file-transfer protocol called the Williamsburg protocol has been added on top of the basic ARINC protocol. It allows for the transfer of as many as 253 data words in a block. The protocol provides flow control, data acknowledgment, and specifies a cyclic-redundancy-check (CRC) polynomial that can be used for error detection. The Williamsburg protocol is used for on-airplane upload of databases, flight-plan information, and software updates. An ARINC 429 bus can operate at two different speeds. The low speed is between 12 and 14.5 kHz and was provided to make ARINC 429 compatible with the older ARINC 419 standard. It is also suitable for general-purpose applications where data latency is not of particular concern. The higher speed is 100 kHz; it is used when there are moderate amounts of data to be transmitted.

The lack of multiple transmitters and its slow speed has made this bus obsolescent. However, as with the 1553, the large number of legacy systems and components mean these old buses will still be with us for many years.

6.4.3.3 SAE High-Speed Data Bus (HSDB)

The AS 2 committee has developed a pair of token-passing networks to supplant the aging 1553. These are a 50-mbit/s bus and a 100-mbit/s ring. The latter is similar to the commercial FDDI network. During the development of requirements for these networks, the committee decided not to include VMS requirements. Therefore, these designs do not include most of the desirable characteristics described above. At one time, there was discussion of putting a warning in the SAE documents saying that these networks were not designed for systems with stringent dependability requirements. The situation now is that it is "caveat emptor" for anyone deciding to try using one of these networks for any highly dependable systems. Neither of these networks has as yet entered service.

6.4.3.4 ARINC 629

The ARINC 629 Periodic-Aperiodic Multi-Transmitter Bus [6.8, 6.9] is intended for use on commercial transport aircraft entering service in the 1990s. It began as the Digital Autonomous Terminal Access Communication (DATAC) bus, which was developed by the Boeing Commercial Airplane Company in an attempt to devise a successor to ARINC 429. The ARINC 429 bus had become a bottleneck for some newer functions because of its relatively low speed. Moreover, the unidirectional nature of the protocol had led to the proliferation of ARINC 429 interfaces on LRUs and therefore to complex wiring harnesses that were comparatively difficult to manufacture and maintain.

Boeing had several key requirements for the new interunit bus. To reduce the amount of wiring between LRUs, Boeing felt it should be a bidirectional bus that could carry traffic from several transmitters. Because there would be multiple transmitters, some means of coordinating access to the bus had to be chosen. Boeing felt autonomous control would be preferable to a master bus controller because centralized functions make a system subject to single-point failures and are difficult to make fault tolerant. As the bus would be shared, the data rate would have to be significantly higher than the 100,000 bits per second provided by the ARINC 429 standard. The bus should have a high level of integrity so that LRUs implementing flight-critical functions could communicate over it. Finally, the new standard should be flexible enough to accommodate different types of LRUs running different types of functions.

The DATAC system Boeing designed is a masterless broadcast bus that employs a carrier sense, multiple-access, clash-avoidance media-access protocol. This access scheme guarantees each terminal has periodic use of the transmission channel at low loads, but under heavier loads the transmission delays can become unpredictable.

Each DATAC terminal is assigned a unique terminal gap (TG) time, which determines the terminal's turn to transmit. Before a terminal can transmit, the bus must have been idle for the terminal's TG. In other words, a terminal must wait for all terminals with smaller TGs to send their messages. If the TGs were the only means of determining the right to transmit, terminals could be "starved" of transmission time. If terminals with smaller TGs always had something to transmit, terminals with larger TGs would never be given the opportunity. To ensure fairness, the protocol also has a transmit interval (TI) and a sync gap (SG). Each terminal is allowed to transmit only once per TI, and a terminal is allowed to transmit only once after it has seen a gap at least as large as a sync gap. As long as the sum of all traffic (plus accumulated TGs) is less than TI, then TI paces the bus. This type of operation is called periodic mode and provides isochronism. The period is equal to TI.

If all offered traffic cannot be transmitted within a TI, the bus is said to be overloaded. The sync gap is larger than the largest TG, which means that each terminal gets exactly one turn to transmit between each SG. This type of operation is called aperiodic mode. The period between SGs varies by bus loading. For VMS systems such as the 777 FBW, the system is designed such that an overload should only occur as short transients at start-up and during fault reconfiguration.

The DATAC system can operate in broadcast mode or in point-to-point (directed) mode at raw speeds as high as 2 mbit/s. At this data rate, the usable data bandwidth is theoretically 100,000 16-bit data words per second, which is 1.6 mbit/s. The flexibility to support a variety of target systems is provided by programmable data-transfer schedules that define the specific data "labels" to be transmitted by a terminal during its turn. A second receive table defines which labels should be copied into local memory whenever they appear on the bus. Terminals can be programmed with more than one transmit schedule to allow for different transmission needs. These tables are stored in nonvolatile memory in each terminal.

In 1986, the Airlines Electronic Engineering Committee (AEEC) formed the Data Bus Subcommittee to develop the ARINC 629 standard from DATAC. The major modification the committee made was to define an alternative to DATAC's access-control protocol. The ARINC 629 specification was approved and released in March 1990.

Each ARINC 629 installation must adhere to one of two access-control protocols: the basic protocol or the combined-mode protocol. The basic protocol is essentially the DATAC protocol. A bus using the basic protocol can operate in either a periodic or an aperiodic mode; when it is operating in the periodic mode, cyclically available data is transmitted cyclically. If the bus is overloaded, it automatically shifts to aperiodic mode. The order of transmission is fixed by the sequence of TI following each (SG), but the frequency with which a given terminal gets a chance to transmit now depends on the load. Conversely, periodic mode fixes the period, but the order of terminal transmissions within that period is not guaranteed. The Boeing 777 uses the basic protocol.

The combined-mode protocol allows simultaneous periodic and aperiodic data transmissions. Under this protocol, the periodicity of selected transmissions is ensured even when the bus is heavily loaded. Data are assigned to one of three categories that have different priorities for bus access. Level 1 messages are guaranteed transmission in every frame. Level 2 and level 3 messages are both sent only if time permits, but Level 2 messages are sent before Level 3 messages.

Both protocols allow for the transmission of either broadcast messages or directed messages. Special provision is also made under both protocols for transmitting bulk data, such as navigation databases. If the bus is utilizing the basic protocol, one of two approaches may be used. On the 777, such bulk transfers will be performed by breaking the data into packets, which will then be transmitted as periodic data according to the primary schedule. The other approach is to allow the terminal to assume an alternative transmit schedule for a bulk transfer. If a large amount of data is being transmitted, the bus may transition to aperiodic mode during the transfer, resuming its periodic transmit schedule when the transfer is complete. If the bus is utilizing the combined-mode protocol, bulk transfer may be handled by structuring the data as a series of single-block aperiodic Level 3 messages.

The ARINC 629 protocol is implemented in a single VLSI circuit, which is made by two different manufacturers. As access to an ARINC 629 bus is autonomously controlled, each terminal must contain its own control information. This information is stored in two programmable "personality PROMs," one for the transmit function and the other for the receive function. The Transmit Personality PROM (XPP) contains information that is used to determine when transmissions should occur and what information should be transmitted. The Receiver Personality PROM (RPP) is used to select only those messages that are intended for the terminal and to monitor the transmitter for babbling and other malfunctions. To work correctly, all the XPPs and RPPs on a bus must have compatible versions. There is nothing in the protocol to guarantee this compatibility. Compatibility enforcement has to be done by some external mechanism.

Whereas ARINC 429 buses are made of shielded wire, which must be cut and stripped to make connections, ARINC 629 can use any of three media: shielded wire, unshielded wire, or fiber-optic cable. Three modes of bus coupling are possible: current mode, voltage mode, and fiber-optic coupling. Current mode, or inductive coupling, has the noteworthy advantage that the wire does not have to be cut to make a connection. The intention was to substantially improve reliability and reduce the effects of electromagnetic interference. However, the inductive couplers are essentially transformers with just a single loop of wire on the bus side. This makes it difficult for a transmitter to induce a strong signal on the bus. To compensate, the coupler needs high-power active electronics, which means running power through the stub and heat sinking the coupler. Similarly, the receiver picks its signal off the bus using the one loop of wire on the bus side of the coupler. The Boeing 777 uses inductive couplers.

6.4.4 Intrabox

6.4.4.1 The Proprietary Past

As opposed to the number of avionics interbox network standards, there are currently only two standard avionics intrabox buses. These standards are relatively recent developments. The intrabox networks used by VMSs have been, to date, purposely built proprietary backplane buses.

6.4.4.2 PI-bus

The PI-bus was jointly developed by IBM, Honeywell, and TRW in partial fulfillment of work being done for VHSIC Phase 2 Submicrometer Technology Development contracts. Work under these contracts also defined Test and Maintenance (T&M) buses that were the foundation of the IEEE 1149 standards. The PI-bus Specification was first released in mid-1985. This bus was subsequently adopted by the SAE AS 2 C committee when first usage was envisioned to be the JIAWG avionics for the F-22 and RAH-66.

The PI-bus is a linear, multidrop, synchronous bus that supports digital message communications among up to 32 modules residing on a single backplane. Messages are transferred datum serial and bit parallel using a datum size of 16 bits (single word) or 32 bits (double word).

The PI-bus uses a master-slave communication protocol that allows the bus master to read data from one slave or write data to any number of slaves in a single message sequence. Messages may be routed to particular modules using either logical or physical addressing. A number of independent messages may be transmitted during a bus master's tenure. The message formats provide a 32-bit virtual address range within each module.

The PI-bus protocol specifies a set of bus state transitions that control the communication sequences and allow the bus to operate in a pipelined manner at the maximum clock rate allowed by the bus signal propagation delay. Master-slave handshaking is provided with a minimal performance penalty by operating the slave modules in synchrony with the master and using bus state look-ahead. This look-ahead feature allows the bus to use full handshaking on each transfer cycle and to do flow control (via a wait control line), even though the operation is pipelined to transfer one datum per clock cycle.

A technique for temporarily suspending low-priority block data transfers to reduce bus acquisition latency for higher priority messages is defined. The bus also has a mechanism for resuming any number of these suspended messages.

Bus mastership may be changed either by direct assignment (token passing) or by priority arbitration. The protocol defines 128 logical levels of message priority and 32 levels of physical priority.

Extensive signal line and sequence error detection capability are incorporated into the bus definition. In addition, an optional single-line error correction capability is specified. The PI-bus can detect and optionally correct any bus line error, including control lines. This error detection/correction covers all bus cycles, including the arbitration cycles. The PI-bus was the first standard backplane bus to have this level of fault detection/correction. (It is now only surpassed by the SAFEbus.)

Bus configurations and modules that conform to the PI-bus standard may be any of the types and classes specified below:

- Type 16 = 16 bit data transfers
- Type 32 = 32 bit data transfers
- Class ED = Error Detecting
- Class EC = Error Correcting

Buses and modules are classified according to their maximum capabilities. Bus sequences are classified according to the type or class of transfer actually used.

All modules and buses must be capable of operating in Type 16, Class ED mode. Type 32 and Type 16 modules are interoperable on a Type 32 bus where the Type 32 modules may communicate using 16- or 32-bit transfers. However, only 16-bit transfers are used whenever a Type 16 module is an active participant. All active modules on a given bus must operate in the same class.

Conceptually, each module connected to a PI-bus consists of a device that performs the application-specific function of the module and a bus interface that implements the PI-bus master-slave communication protocol. The device portion of each module is modeled as a virtual memory space with a 32-bit address range. The bus interface is modeled as a separate memory space with an 8-bit data link register address range. A separate 8-bit virtual address called the slave ID is used by the bus master to select one or more modules to participate in a particular communications sequence as slave(s).

The Physical Layer of the PI-bus uses Backplane Transceiver Logic (BTL) [6.]. This driver logic has several advantages over the older TTL drivers, including lower capacitance and higher speed. Other new backplane standards, including Futurebus+ and SAFEbus, also specify it. The PI-bus is designed for incident-wave switching (does not wait for any transmission line reflections).

The clock network that paces the PI-bus is not considered part of the bus itself. The clock network probably would have to be a star network rather than a bus to minimize the clock skew among the points where it enters each module.

6.4.4.3 ARINC 659 (SAFEbus)

The SAFEbus [6.11] protocol is driven by sequences of commands stored in the table memory of the Bus Interface Unit (BIU). Each command corresponds to a single message on the bus. The command indicates whether the BIU should transmit, receive, or ignore the message. The BIUs are synchronized so that at any given time, all BIUs are at equivalent points in their respective tables. Mechanisms are provided to quickly regain synchronization should it ever be lost. The tables also contain the local address of the data to be transmitted or received. The commands in each BIU's table are organized into loops (or frames) of equal duration.

All transmissions on SAFEbus are 2-bit serial. Compared to other parallel backplane buses, this drastically reduces the total module pin count, increasing the system's inherent reliability. The bus time is divided into a set of "windows," each of which contains a message of from 32 to 8,192 bits. The windows are separated by a small, fixed gap time, which is approximately the time it takes to transmit 2 bits of data. Messages that are to be transmitted or have been received over the backplane are placed in buffers in intermodule memories. This organization permits a simple host interface, because the hosts view SAFEbus essentially as a memory-mapped peripheral.

One of the benefits of the table-driven protocol is extremely high efficiency. VMS applications typically generate short backplane messages, and most serial protocols perform poorly when messages are short. Efficiencies of between 10 and 30% are typical. The SAFEbus protocol, on the other hand, is over 88% efficient for a continuous stream of 32-bit messages. Because buffer addresses are kept in tables, they do not need to be transmitted on the bus. The use of transmit and receive commands in the individual tables eliminates the need to send source or destination module addresses. Because transmissions are scheduled, no transmission time is consumed arbitrating between contending BIUs. Except for the intermessage gap and the occasional synchronization message, all bit times contain data. Thus, the 60-mbit/s SAFEbus on a 2-ft-long backplane has a usable throughput that ranges from 51 to 59.97 mbit/s. The backplane can be 2-bit serial rather than wide-parallel because the protocol is so efficient.

SAFEbus actually consists of dual self-checking buses (SCBs), and each SCB is itself composed of two buses. The interface logic, including the BIUs, is also duplicated. One of the BIUs transmits data on one of the data lines in an SCB, and its partner transmits on the other data line. The data on the two lines are compared at the receiver. If a miscompare occurs, the data are discarded instead of being written into intermodule memory. The receiving circuitry in the transmitting module also checks what is actually put on the bus for errors. Finally, self-checking ensures that a babbling module will be detected and will remove itself from the bus, as there is little likelihood that both sets of interface logic will fail in the same way at the same time. In general, the SCBs provide error-detection coverage that exceeds that provided by CRC codes, and they do so without consuming transmission time.

The second set of buses provides immediate error correction for single-SCB transient errors. This also makes it more likely that the functions in the cabinet will remain available despite failures. SAFEbus is fail-operational/fail-passive; if one set of buses fails, the cabinet remains in operation; if the second fails, the cabinet goes quiet.

For the physical layer, the SAFEbus protocol uses Backplane Transceiver Logic (BTL). The new logic has several advantages over the older TTL drivers, including lower capacitance and higher speed. Other new backplane standards, including Futurebus+ and PI-bus, also specify it.

The determinism of this design warrants more detailed examination, as no other backplane protocol provides it. Common weaknesses in backplane designs include the use of FIFOs, the inclusion of destination addresses in messages, and the use of media-access protocols that involve arbitration.

FIFOs, which are often used for communication buffers, can be a space- and time-partitioning problem. A malfunctioning function can fill a FIFO, preventing other functions from gaining access. If a system has FIFOs, the only way to guarantee partitioning is to perform extensive worst-case traffic analysis.

Any protocol that includes a destination memory address in a message is a space-partitioning problem. It is extremely difficult to verify correct address usage in a partitioned multiprocessor. To ensure correct usage, the BIU would have to duplicate the processor's memory-management function.

Finally, any protocol that uses arbitration cannot be made time-deterministic. Arbitration is meant to ensure that when two modules contend for the bus, the one with the highest priority request is granted access. However, minor jitter in the execution of functions can change which modules contend for the bus. As a result, the order in which the modules obtain access can vary from frame to frame.

SAFEbus achieves both time and space determinism by placing all message locations and bus-timing information in the table memories. Fixed mapping of messages to unique locations in the intermodule memory, which is protected by memory-mapping hardware in the host, guarantees space determinism. In addition, all of the control information in the intermodule memories is automatically modified under BIU control, a precaution that guarantees its consistency.

To make the system even more predictable, the execution of the software in the processing modules is synchronized with the execution of the commands in the bus table. Thus, the application software is at the same point during the same bus transmission window in every frame. One benefit is that message latencies are reduced; results can be scheduled to be transmitted just after they are generated, and data can be brought in from the I/O modules just before they are needed. A second benefit is that there is less latency jitter on cabinet outputs, which means that SAFEbus can be used in tighter control loops. A third benefit is that double buffering is rarely necessary because it is possible to schedule the transmission of a data block for a time when it is known that the function software will not be accessing or modifying it. The elimination of double buffers means that intermodule memories can be smaller and memory access faster.

The synchronization of the bus schedule and the application software's execution is guaranteed by embedding interrupt commands in the SAFEbus tables. On receiving an interrupt, the processor's operating system shifts to another application program. The interrupts take the place of the hardware timer other real-time executives employ. Because the interrupts are not transmitted over the bus, each BIU can have a different interrupt schedule. This allows the system integrator to fine-tune the operation of each core processor or I/O module.

The synchronization of the bus and software also assists debugging and validation. First, because SAFEbus is deterministic, a function experiences the same system timing whether the cabinet is fully populated or nearly empty. Second, because each processor is synchronized to the bus, the processors are implicitly synchronized to each other. Thus, any timing conflicts between functions running in different cores will be quickly exposed, making the system simpler to debug. In asynchronously scheduled multiprocessor systems, such timing problems show up as intermittent failures, which can be very costly to track down and make it impossible to validate the system. Third, whenever the system is stopped or single-stepped, it passes through a succession of clearly defined states. The clear relationships between the application programs, which are defined by the SAFEbus table, make it easier to trace behavior.

SAFEbus data messages have been designed to support the requirement of high efficiency. Because the protocol is table-driven, messages contain only data and not address and control information as well. There are two data-message types: basic and master/shadow. The basic message structure has been chosen to maximize the efficiency of data transmissions. The master/shadow structure supports data transfers by redundant functions.

Basic messages have a simple structure. Each message consists of a string of 32-bit data words followed by an intermessage gap of two bit times. Information other protocols send in message headers is stored in the SAFEbus tables instead. There is no CRC code in the message, because the self-checking buses provide the required error-detection coverage. Thus a single 32-bit ARINC word occupies only 18 bit times on the two-bit-wide bus, which means the transmission is 88% efficient. The protocol exceeds 99% efficiency for messages longer than a dozen 32-bit words.

The master/shadow mechanism allows modules or applications to be reconfigured or spared without disturbing the traffic pattern on the bus. Master/shadow windows are identified by a field in the table command. As many as four transmitters can be assigned to one master/shadow window. Time-slot arbitration determines which of the transmitters actually gets control of the window. If the master is alive and has fresh data to send, it starts transmitting at the beginning of the window. The first shadow begins transmitting "delta" bit times into the window, but only if the master did not use its opportunity to transmit. The second shadow begins transmitting two delta bit times into the window, but only if the master and the first shadow did not use their opportunities to transmit. The third shadow is similar, waiting three delta bit times. Delta is a programmable value that is typically set at three or four bit times. The selected value depends on the propagation characteristics of the backplane.

Time-slot arbitration could reintroduce nondeterminism, but strict measures have been taken to eliminate this danger. First, extra bit times in the window and a restriction on the size of the message guarantee that message transmission will be completed within the assigned time window, no matter what happens during arbitration. Second, the time window remains the same size, no matter which transmitter “wins” the arbitration. Third, recipients of a master/shadow message always place the data in the same memory location, no matter which transmitter wins the arbitration. Fourth, all recipients are notified of the arrival of a message at a fixed time after the beginning of the window, no matter what happened during arbitration. Fifth, delta can be made large enough to guarantee that the candidate transmitters will never mistake a busy bus for an idle one and begin transmitting in error.

The SAFEbus synchronization messages have been designed to support the requirements for integrity and time determinism. Cabinet synchronization is guaranteed in the face of any reasonable failure scenario, and synchronization does not require any centralized resource that could diminish the system's integrity. Three types of synchronization message are provided to bring the BIUs in step under three circumstances: at power up, when a module is "lost," and during normal operation.

The initial sync message is used to start up an inactive SAFEbus. It would be regenerated automatically if a pathological cabinet-wide loss of synchronization should ever occur. The long resync message contains sufficient information to allow a "lost" module to regain synchronization with an active bus. The short resync message is used to maintain bit-level synchronization of all BIUs on the backplane.

After any synchronization message is received, all BIUs are synchronized to within one bit time. Subsequently, their oscillators could eventually drift one more bit time and thus close up the intermessage gap. Short resync messages are programmed into the command tables frequently enough to prevent gap closure.

To provide additional fault tolerance, all BIUs transmit the sync-pulse portion of every synchronization message. The multiple sync pulses are combined into a single pulse by the open-collector BTL drivers. Also, any module can originate an initial sync pulse. Because the synchronization mechanism is decentralized, no particular module must be operational to start up the backplane or to maintain synchronization.

To improve error-detection coverage, data on the four SAFEbus serial lines are encoded in four different ways. Data on Bus Ax have normal polarity. Data on bus Bx are inverted. On bus Ay every other bit is toggled, starting with the second bit. Bus By is the inverse of bus Ay. This encoding scheme allows detection of bus shorts or transient upsets that affect several data lines simultaneously. It also allows quick detection of bus collisions caused by malfunctioning BIUs. Because bus lines are "wired OR," if a BIU pair malfunctions and tries to transmit at the same time as another BIU pair, illegal encodings appear within two bit times. An additional virtue of this encoding scheme is that power consumption is independent of the data being transmitted. Two bus lines are always high and two are always low; when the data change, two of the buses change state and two do not. Because power consumption is constant, the power supply does not have to be designed for a worst-case data pattern. This encoding also has other benefits similar to differential signal lines.

The information stored in the EEPROM table memories is the heart of the SAFEbus protocol. The table memory is divided into three areas: the resynchronization jump table, the command sequence area, and the BIU configuration area.

The sync code in a long resync message serves as an index to the resynchronization jump table, which contains pointers to memory locations in the command sequence area. In this way, a BIU can regain synchronization within one window time after receiving a valid long resync message.

The BIU configuration area contains information that customizes the operation of the BIU. This includes a "slot" number, which indicates which module's table the memory holds, the table version, and the table CRC. The CRC allows the BIU to verify the integrity of the table at power-up. Finally, this area contains data about customization options such as memory speeds, host interface characteristics, and the SAFEbus timer increment rate.

The remainder of the table memory is used for the cyclic command sequences that define the data transfers that occur during a frame. More than one command sequence, or frame, can be present in the table. The alternative frames provide for different system modes, such as system initialization or ground checkout. The mechanism for switching between frames requires that operating-system software generate explicit commands that must then "cooperate" with BIU table commands to effect the switch. Because the BIU's cooperation is required, it can prevent errant operating-system software from switching between frames inappropriately.

6.4.4.4 Interbox Comparisons

Table 6-2 compares the SAFEbus and PI-bus, as well as the VMEbus and Futurebus+. While the latter two buses were not developed for avionics and have characteristics that are not a good match for those desired for a VMS, these buses are sometimes suggested for use in avionics.

Table 6-2. Backplane Bus Comparison

	SAFEbus®	PI-bus	VMEbus	Futurebus+
Data bits [width option]	2	16 [32]	16 [32]	64 [128, 256]
Signal and clock lines				
• No fault tolerance	3	30 [47]	67 [107]	122 [186, 256]
• SED	6	30 [47]	134 [214]	244 [372, 628]
• SECDED	12	43 [59]	201 [321]	366 [588, 942]
• DECTED	12	73 [106]	268 [428]	488 [744, 1256]
32-bit message throughput (Mbyte/s)	6.66	5.0 [10.00] (SAE) 0.9 [1.50] (IBM)	12.9	15.5
Throughput per pin (SECDED)	0.83 (30 MHz)	0.02 [0.03] (IBM)	0.06	0.04
Arbitration time (ns)	Zero	800	136,900	89,500
Address space (byte)	Unlimited	8 x 2^{128}	64 x 2^{24}	1 x 2^{64}
Multiplexed address	No	Yes	Yes	Yes
Clocks or strobes	Source clocked	Synchronous	Asynchronous	Asynchronous, source clocked
Partitioning enforcement				
• Memory protection	Yes	No	No	No
• Time determinism	Ordinal and cardinal	Ordinal (with tenure pass)	None	None
Built-in fault tolerance				
• Bus lines, pins	Yes	Yes	No	No
• BIUs, drivers	Yes	No	No	No
Broadcast capability	Yes	Yes	No	Yes
Driver technology	BTL	BTL	TTL	BTL
Live insertion	Yes	Allowed	No	Yes
Debugging	Easy	Moderate	Very difficult	Very difficult
Built-in *system* debug features	Yes	No	No	No
Designed for *system* validation	Yes	No	No	No

Notes:

SED = Single Error Detection (any bus line or pin).
SECDED = Single Error Correction, Double Error Detection (any line or pin, nonidentical faults).
DECTED = Double Error Correction, Triple Error Detection (any line or pin, nonidentical faults).
SAE = Best theoretical performance with SAE/J!AWG 12.5-MHz PI-bus.
Address space = number of spaces x number of bytes per space.
Ordinal = Order of events guaranteed.
Cardinal = Time between events guaranteed.
Asynchronous buses are difficult to test and diagnose; conditions are not repeatable.
Asynchronous buses are less well behaved in the face of metastability.
Source clocking is the fastest method.
BTL is superior to TTL for high-speed backplane applications.

6.5 ELECTRONIC PACKAGING

The desire to achieve faster, smaller, higher capacity computational systems has resulted in new concepts to package electronics. Mostly to create smaller packages, the newest Multi-Chip-Modules (MCMs) help create impressive speed advantages through enhanced communication between chips packed tightly in two and three dimensions.

As illustrated in Figure 6-3, increasing sophistication in packages comes at a cost. As with other advances in electronics the costs will eventually be affordable for many applications. Using MCMs and chip-on-board techniques, we will see yet another impressive set of computer capability advances, shown in Figure 6-4.

Perhaps the biggest unresolved issue for aerospace embedded use, however, will be cooling, as these new systems will require much improved heat removal per volume than previous concepts.

6.6 POWER DISTRIBUTION

A source of extremely dependable power is increasingly becoming a requirement for aircraft systems. During total or even partial power failure, the subsequent loss of data or power can be catastrophic for flight-critical applications such as flight control.

A general block diagram of a representative power control system is shown in Figure 6-5. The power control and conditioning (PC&C) module after the power generator provides the power quality functions such as voltage and frequency control. A feeder wire then furnishes required cross connections to the power buses, which provide power transmission and distribution. The PC&C function at the load is performed by a solid-state power controller (SSPC). The function of the SSPC is to control power to the load and give status information. The IVMS can interface with the power system at this level. On/off information is transmitted from an IVMS to control the load and, in turn, status information can be returned from the SSPC to a central health monitoring system.

Storage batteries are a special concern in a power system. The batteries are required to provide backup emergency power. This backup power must supply flight-critical systems in the event of a power system failure. Issues are the ability to assess and ensure the charge of the batteries so that they are always available if required.

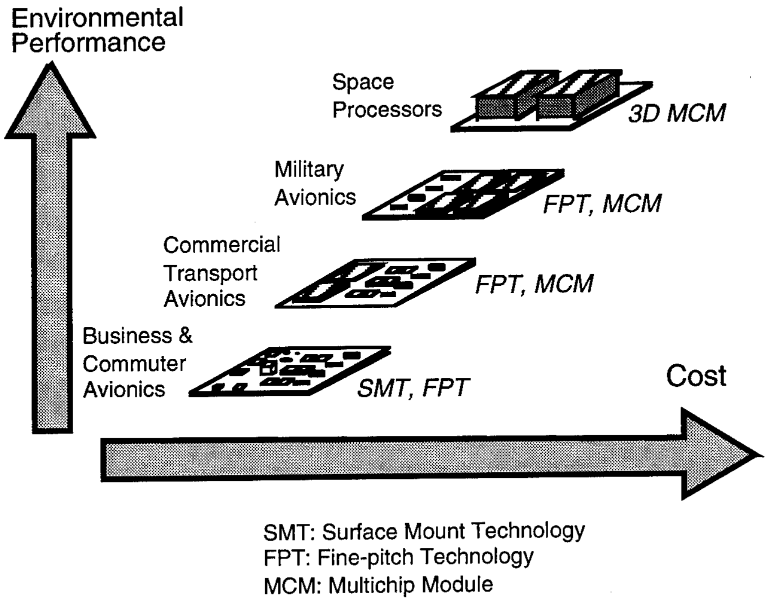

Environmental Performance
Space Processors
3D MCM
Military Avionics
FPT, MCM
Commercial Transport Avionics
FPT, MCM
Business & Commuter Avionics
SMT, FPT
Cost
SMT: Surface Mount Technology
FPT: Fine-pitch Technology
MCM: Multichip Module

Figure 6-3. Packaging Options for Embedded Systems

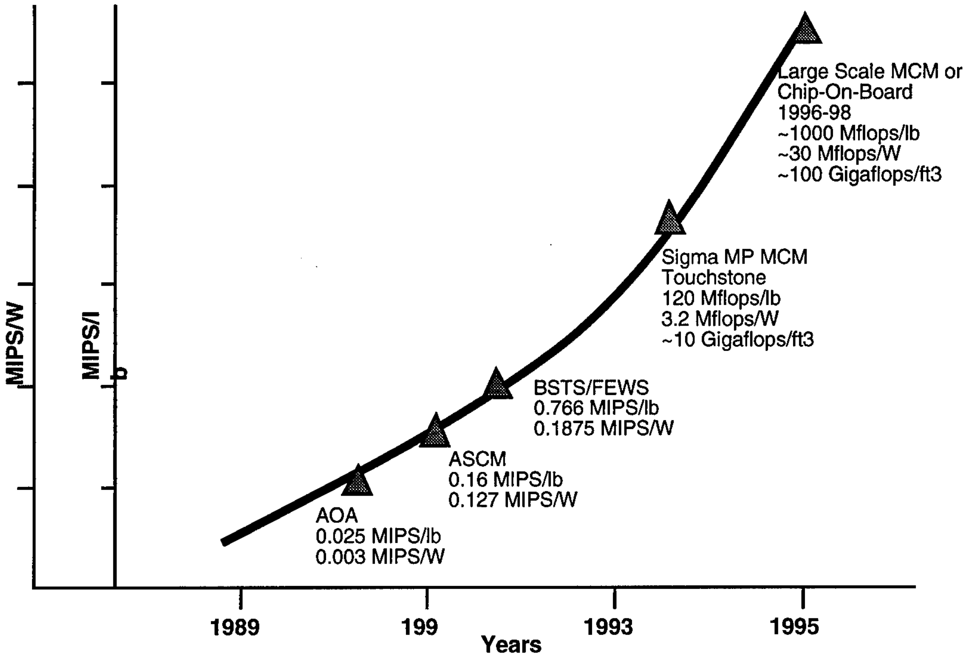


Figure 6-4. Embedded COTS Supercomputers will Extend Previous Embedded Performance

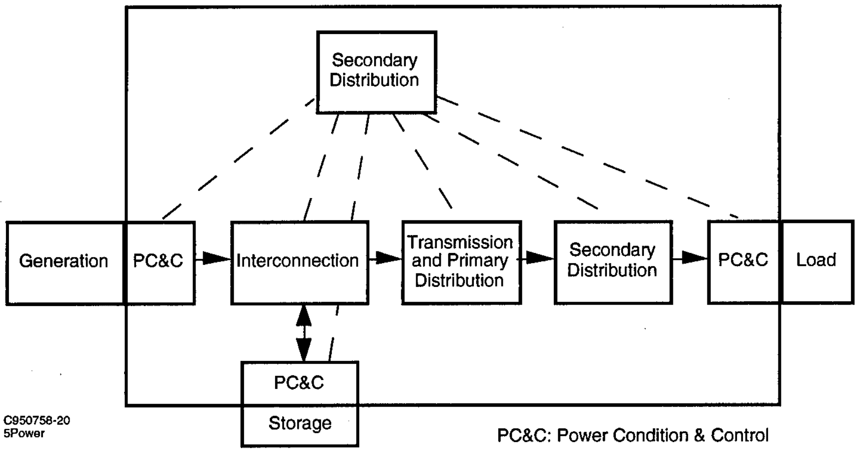


Figure 6-5. Power Control System

There is a trend in computing systems to decrease the operating voltage of the electronics. This results in a reduction in the signal-to-noise ratio and places more demands on the secondary net isolation and environmental protection procedures.

There is also a trend in power generation and distribution systems to utilize 270 VDC. The incentive for such systems is the reduction in aircraft weight due to the reduction in copper for carrying the reduced currents. However, more care is required in voltage isolation and protection.

It is possible to increase the reliability of a power distribution system by introducing fault-tolerant techniques such as parallel redundancy. However, in many situations, this solution can introduce new difficulties in thermal management, size, efficiency, and cost. The next section describes some of the pitfalls in using parallel redundancy with cross-strapping.

6.6.1 Power Cross-Strapping

Many VMS designers try to cross-strap sources of electrical power in a misguided attempt to increase system dependability. This cross-strapping may have been valuable when electronic systems were not flight critical. When that was the case, availability was the main concern. Now that flight-critical electronics are used, integrity is equally important. Cross-strapping is detrimental to integrity. To explain the pitfalls of this design idea, this section describes a simple example. Figure 6-6A depicts a hypothetical avionics system consisting of two power supplies and two sets of electronics.

The two power supplies and the two sets of electronics are given the same 10^{-4} probability of failure to make the mathematics simple. If the total system consisted of only the two power supplies and the two sets of electronics, and if each power supply were only connected to the set of electronics directly to its right, then the first equation line gives the probability of total system failure. This is the probability that:

- Both power supplies fail = $(10^{-4})^2$ or
- Both sets of electronics fail = $(10^{-4})^2$ or
- One power supply fails and the opposite set of electronics fails (which can happen two ways) = $2*(10^{-4})^2$.

The probability of total failure is then: $(10^{-4})^2 + (10^{-4})^2 + 2*(10^{-4})^2 = 4*(10^{-8})$.

Designers are always tempted to remove the third term of the equation by cross-strapping the power supplies. With DC power supplies, this is done with diodes such as those shown in Figure 6-6B. However, it is immediately apparent that a short to ground in either set of electronics becomes a single point of system failure if only diodes are used. Therefore, fuses are added as in Figure 6-6C. This diode cross connection creates other single points of failure. For example, if one of the power supplies produces an overvoltage, both sets of electronics are affected. Therefore, the overvoltage protection is added at the source. A known overvoltage scenario is if the feed valve on a bleed-air APU sticks open, produces an overvoltage, which passes through a diode cross connection, which triggers overvoltage protection in electronics on all flight control channels, thus shutting them all down and losing all control. Another problem is spikes on the power line. These spikes can be narrow enough to not trip the fuses but have enough amplitude to destroy electronics. Therefore, spike suppressers are added. The electronics also can produce RF noise on the power lines that cause additional malfunctions. This RF can be coupled through the cross connection and cause the opposite channel to malfunction. Now RF filters have to be added. The accretion of all this additional protection circuitry is shown in Figure 6-6D.

When all of these extra hardware elements are added, the equation for probability of total system failure is:

$$(10^{-4})^2 + (10^{-4})^2 + ? + ?? << 4*(10^{-8})$$

The first two terms have not changed. The third term is replaced by ? and ??, which represent, respectively, the probability that some component in all this cross-connection hardware will cause a single-point failure and the probability that there is still an unknown single-point failure mechanism due to the coupling of failure effects from one channel to the other.

For all this added complexity to have any merit, the new total system failure probability must be much less than the original $4*(10^{-8})$. Therefore, the two unknown single-point failure classes introduced by the cross connection must have a combined probability of occurrence of much less than $2*(10^{-8})$. This means that the FMEA and V&V processes must prove that the combined probability of all possible of these single-point failures is much less than $2*(10^{-8})$. This is far beyond any current FMEA or V&V capability.

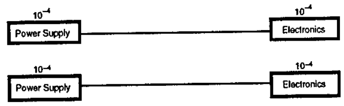


Figure 6-6A. Dual Power Supplies

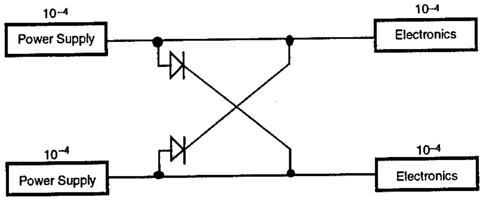


Figure 6-6B. Dual Power Supplies with Diode Protection

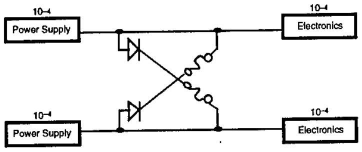


Figure 6-6C. Fuse Protection Added to Power Supplies

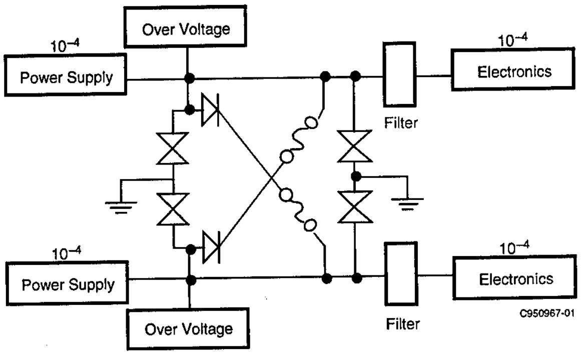


Figure 6-6D. RF Filters Added to Power Supplies

The bottom line then is: until giant strides are made in FMEA and V&V technology, it is never a good idea to cross connect power supplies in this manner for flight-critical systems.

6.6.3 Solid-State Power Controllers

The SSPC, as described previously, provides the interface between the load and IVMS. These SSPCs have TTL/CMS-compatible inputs and outputs that allow control from and real-time monitoring by the CPU. For this reason, they have to be rugged and environmentally sound. Two types of solid-state devices can be used in the SSPC.

Normally, a bipolar transistor has a safe operating area defined by a set of current-voltage limits. If this is exceeded, local hot spots occur. These hot spots conduct currents more readily than adjacent areas and tend to become hotter. This thermal runaway leads to the eventual destruction of the device.

Power MOSFETs have the opposite characteristic; a local hot spot will “steer” current away as its resistance increases. This eliminates hot spots and results in even current sharing across the device. The inherent advantage of this is that the entire MOSFET has to exceed its temperature limitations before damage results; this property makes the power MOSFET more rugged than bipolar when used for power switching. Another advantage of MOSFET is that they are stable across the whole military temperature range (-55° to + 125° C). This device is the choice for the SSPC function.

The main features of SSPC are:

- It2 (energy) protection,
- Isolated control circuits,
- Status outputs,
- Built-in test,
- Low power dissipation,
- Solid-state reliability.

6.7 PARTITIONING TECHNOLOGY

The most important group of technologies for enabling the integration of a VMS are those that provide robust partitioning. A partition is a unit of protection in the same way that a process or a task is a unit of execution. A partition may contain several processes or tasks; but, they all must belong to the same function and have the same level of criticality. The granularity of partition size (number of tasks, amount of code, number of variables, etc.) is a tradeoff. The smaller the partition is, the easier it is to do verification. However, smaller partitions mean more partitions, each of which incurs an overhead for providing its protection. This protection must cover both space and time partitioning as described in section 2.11.

A robust partitioning mechanism is one which supplies sufficient protection so that each partition is immune to the misbehavior of it neighboring partitions, regardless of their level of criticality. The latter requirement means that a robust partitioning mechanism must expect the most devastating misbehavior possible out of each partition. The source of failures that robust partitioning must protect against are design failures in the software and in the hardware that is unique to a particular function. Failures in the common underlying hardware happen regardless of whether the system is integrated or not; and therefore, do not pose any additional partitioning concerns for an integrated system. The terminology for robust partitioning has been codified in commercial avionics by the ARINC 651 document. The Boeing 777 Aircraft Information Management System (AIMS) is the first system to enter service which provides robust partitioning.

6.7.1 Space Partitioning Via Memory-Management Units

The most common piece of hardware used in a robust partitioning mechanism is a memory-management unit (MMU). The MMU provides space partitioning by restricting access to areas of a processor's memory address space on a per task or process basis. An MMU restricts the address ranges that may be accessed and the type of access (e.g. read, write, execute) for each range. Restrictions are enforced by the MMU via tables which define a mapping between a processor's virtual address used by its machine instructions and the physical address of the devices connected to the processor.

For a robust partitioned system, there is a unique map for each partition. No more than one partition may have write access to any address range. A partition may be allowed to read any memory address, as long as that memory read has no side effects. A common example of memory reads which cause side effects are interrupt registers (or similar input device registers) which automatically clear when read. For memory areas that have side effects, only one partition may have read access. A partition may have read access to a memory area that another partition has write access to, as long as timing of these access are carefully controlled. These shared read/write accesses MUST be controlled by a time partitioning mechanism (see section 6.9.3). Each task or process within a partition may have an MMU map which restricts it to some subset of its partition's map.

The MMU's tables must be managed by a specially privileged piece of software (usually called the operating system, executive, or kernel) which has have a robustness and criticality at least as high as the most critical partitions that the MMU protects. Most modern processors that are candidates for an IVMS have hardware which restricts control of the MMU to just this one specially privileged piece of software. Whenever the time partitioning mechanism determines that the partition currently running on the processor needs to be replaced by another partition, this special MMU managing software disables the current partition's map and loads in the map for the new partition. In general, the maps are all predefined at design time. The MMU managing software just needs to make sure that the correct MMU tables are loaded for each partition switch.

By using memory-mapped I/O, the MMU's protection can be extended to cover non-shared I/O devices. Without this MMU-partitioned memory-mapped I/O, software with special hardware supported privileges would have to intercept all non-shared I/O to ensure space partitioning. Shared I/O falls into the domain of time partitioning, which will always need to be intercepted by specially privileged software and/or hardware. This intermediary must be as robust as the rest of the partitioning mechanism(s).

6.7.2 MMU Inadequacies

6.7.2.1 Latent Faults

A fault in an MMU can remain latent (dormant) for a long period of time. An MMU fault that would allow an illegal access won't become a failure until some other (e.g. software) design fault causes that illegal access attempt. This problem is the same as the "stuck at good" problems common to all fault detection mechanisms.

The two most often used ways of combating these problems are replication and scrubbing. In this case, replication would use two or more MMUs which must agree before an access is allowed. This agreement can take many forms (e.g. voting with a set threshold) depending on system design requirements and constraints. Just the MMUs could be replicated or the whole protected memory could be replicated (each with it own independent MMU) and the memories' outputs could be voted. As with any redundancy scheme, replication doesn't prevent failures; it only reduces the probability of an undetected fault.

The other method of combating latent "stuck at good" failures is scrubbing. There are two ways of doing scrubbing: (1) causing known failures and checking that the mechanism handles them correctly, and (2) testing that the structure of the mechanism is still intact. The former can be implemented by a partition that tries all known illegal memory accesses. This is sometimes called a "rogue partition". The structural testing tries to determine if the mechanism's logic components and interconnections are still functioning properly. The idea is that if all the components and interconnections are good, the correct operation of the mechanism can be inferred without periodically exhaustively testing all possible faulty inputs. The IEEE 1149.1 test bus, now being added to many new integrated circuits (ICs), can be effectively used for structural testing. Built in self test (BIST) logic is also often used to efficiently test internal IC structure.

The 777 AIMS uses a combination of all of these techniques (MMU replication, rogue partition, and structural testing via IEEE 1149.1 scan and specialized BIST).

6.7.2.2 Multiple Ports

MMUs are usually adequate to prevent corruption on simple uni-processors. For systems with multiple tightly coupled processors (processors sharing the same address space) running separate partitions and/or for a processor with direct memory access (DMA) devices which are partition controlled, a simple MMU is insufficient. This is because these situations use multi-port memories. A multi-port memory can be read and/or written by more than one device. Putting an MMU on only one port protects only that one port; which is obviously insufficient.

The two most common ways that tightly coupled processors implement their multi-port memories are: (1) each processor has a memory for which the processor has a dedicated port and all other processors share another port (usually via a backplane bus), and (2) a "global" memory not located with any particular processor is accessed via a shared bus or multiple private links.

For most implementations of (1), the local processor's port into its memory is protected by an MMU but the other port (normally connected to a backplane bus) is not protected. Most implementations of (2) have no protection on the "global" memory. The only protection in these cases is to require an MMU between all processors (and DMAs) and the buses or links to the multi-port memories. This is problematic because:

- Processors without an MMU cannot be allowed.
- Sometimes buses/links do address translations which confuse the MMU mappings.
- All the MMU maps must be coordinated, which is a logistical problem at design time and can be a logistical nightmare at run time (particularly for dynamic memory).
- One MMU fault can corrupt the memory of many processors.

DMA devices typically do not use MMUs. It is difficult to add an MMU to a DMA device because they typically don't have enough "intelligence" to manage MMU page tables.

A solution to the multi-port problem is to use a backplane bus which does not use addresses. Without an address, data cannot be placed in the wrong address. See section 6.9.4 for a description of how this can be done.

6.7.3 Time Partitioning

Time partitioning must be driven by a time source that cannot be affected by the partitions it is protecting. Typically, this is done with a real time clock (RTC) interrupt coupled to a robust privileged interrupt handler and executive. Other partitions must not be allowed to interfere with the clock source or the interrupt mechanism. In particular, the RTC interrupt must have the highest priority (no other source can interrupt it); other interrupts must not block it (requires interrupt nesting or the RTC must be the only interrupt); and, instructions must not block it (puts maximum limit on worst case instruction execution time, including: exceptions, page faults, cache misses, etc.).

At each partition tick (RTC interrupt which signals a partition change), the partition executive must suspend the current partition and resume the partition which is scheduled to run next. This context switch must sanitize all shared system resources such that the partition which gets suspended cannot leave any state which may effect the partition being resumed. This is becoming more difficult to do as processors increase their "hidden state" (state which is not directly visible or controllable by software) to increase performance. Modern processors have not only data and instruction caches but also branch target caches, translation look-aside buffers (TLBs), various pipelines, register renaming maps, speculative execution partial results, internal power up/down section switches, etc. The context switch must also restore all visible state of the partition being resumed, including exception conditions if this partition had been suspended in the middle of exception handling.

The scheduling of partition ticks can and should done so as to eliminate critical region considerations for memory that is written by one partition and read by another. That is, the software should be scheduled such that this type of shared memory is not being read or written when a partition tick occurs. When this type of scheduling is used, the software doesn't have to implement the semaphores normally required for this type of shared memory inter-task communication.

Time partitioning becomes difficult when the system includes multiple processors which share resources. In this case, the RTCs must be synchronized with a maximum skew less than the minimum context switch time. When the shared resource is used for a large number of small time intervals (e.g. backplane bus transfers) the RTC skews and the skews between access requests must be much less then the small usage time interval. See the following section for an explanatory scenario.

6.7.4 Backplane

A very common form of avionics integration places several processors and/or I/O module/cards on a common backplane bus. While this bus forms the backbone of the integration, it can present the biggest partitioning problems. Common weaknesses in backplane bus designs include the use of FIFOs, the inclusion of destination addresses in messages and use of media-access protocols that involve arbitration.

FIFOs, which are often used for communications buffers, can be a space- and time-partitioning problem. A malfunctioning partition can fill a FIFO, preventing other partitions from gaining access. If a system has FIFOs, the only way to guarantee partitioning is to perform extensive worst-case traffic analysis. The design also must ensure that the worst case analyzed is not exceeded in operation.

Any bus protocol that includes a destination memory address in a message is a space-partitioning problem. It is extremely difficult to verify correct address usage in a partitioned multiprocessor. To ensure correct usage, the bus interface unit (BIU) would have to duplicate the processor's memory-management function. In addition, there would have to be a coordination mechanism which made sure that the BIU's MMU pages were compatible with that of its processor and that of all the other BIUs on the bus.

A solution to this problem is to use a backplane bus which does not use addresses. Without an address, data cannot be placed in the wrong address. Instead of address, data can be identified by time order of transmission. The ARINC 659 (SAFEbus) described in 6.4.4.3 uses such a mechanism.

Any protocol that uses contention arbitration cannot support robust partitioning. Contention arbitration is meant to ensure that when two modules contend for the bus, the one with the highest priority request is granted access. But minor jitter in the execution of partitions can change which modules contend for the bus at any instant. As a result, the timing and order in which the modules obtain access can vary randomly. For example, suppose there are two processors on a bus with one high and one low priority. If the two always happen to request bus usage at exactly the same time, the low priority will always have to wait; thus, doubling the time it takes for each of its bus accesses. If the low priority processor were to request the bus just before the high priority processor, the low priority processor won't have to wait at all, but the high priority will have to wait every time (assuming no preemption). The time difference between "exactly at the same time" and "a little bit earlier" can be infinitely small. In fact, it typically can be zero, where the winner is determined by the totally random settling of a metastable arbitration flip-flop.

The bus access protocols which have this problem include most of today's common protocols, e.g. anything with priority (most backplane busses), carrier sense multiple access (CSMA), token passing.

The bus access protocols which do not have this problem include: command/response, pure time division multiple access (TDMA), and table driven proportional access (TDPA). Command/response and pure TDMA have a single point of failure in the central controller and clock, respectively, for which fault tolerance must be added it they are to be used for an IVMS.

6.7.5 Individual Processors

An alternative to providing robust partitioning within a processor is to assign a single partition to each processor and eliminate any shared memory or data paths. This is the opposite of logical integration. However, with the continuing decrease in size and increase in speed of processors, a physically integrated but logically separated system may become the most cost effective architecture.

6.8 SYSTEM HEALTH MANAGEMENT

An important trend for safety, reliability, and fault-tolerant improvements at lower cost results from advances in techniques and hardware for system health management. The state of the art has evolved over the past 10 years from some simple added sensor modules to some exotic new schemes for assessing structural health of aerospace vehicles.

One basic premise for system health management is the creation of *condition-based* definition maintenance instead of *time-based* maintenance. This results in numerous benefits:

- *Cost*—Reduced life-cycle cost through reduction or elimination of inspections or replacement of parts that are serviceable. This also allows extended usage of aging systems.
- *Safety*—Impending or premature failures are diagnosed in time for maintenance action. This relieves many conditions that have required instantaneous reaction to hard failure occurrence in the past. This anticipation should also alleviate many soft failure conditions that have been difficult for failure management systems to cover.

Figure 6-7 shows the self-contained electronics and sensors associated with a typical health management function.

Features of a unit like this are:

- Numerous low-cost sensor possibilities: temperature, humidity, vibration, pressure.
- Self-contained processors, memory, digital interfaces, and power.
- Low cost: $50-$500 depending upon military specifications.

The future also includes some attention to corrosion health management. This is particularly critical because we are facing extended use of weapon systems, particularly aircraft platforms, well beyond the original life expectancies. One concept now in research involves a "smart rivet" that contains chemical sensors to detect corrosion. Shown in Figure 6-8, this concept, along with emerging low-cost acoustic sensors, would provide continuous monitoring of corrosion and stress for critical vehicle areas. Such techniques would provide a more thorough inspection of aircraft at potentially lower costs than the extensive off-line inspection currently required.

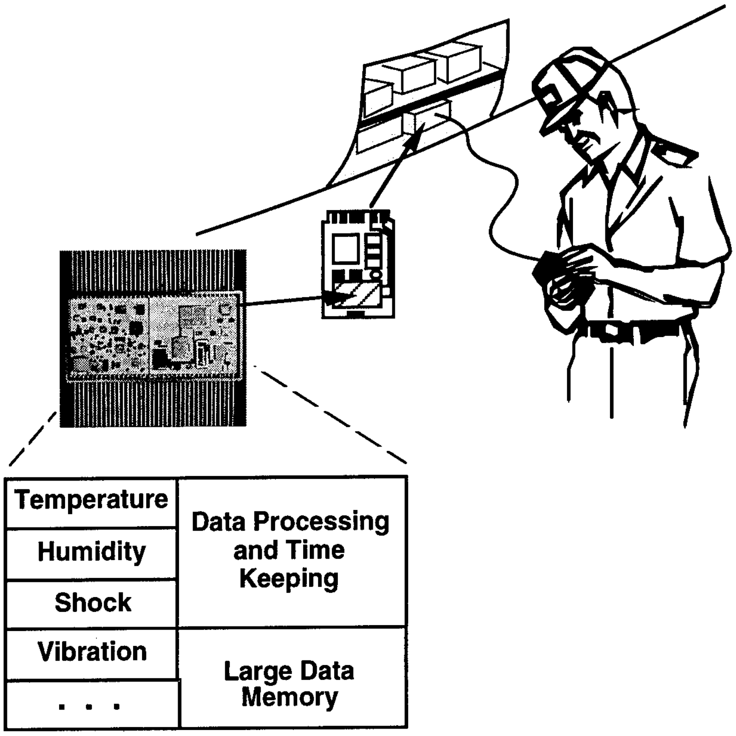


Figure 6-7. System Health Management

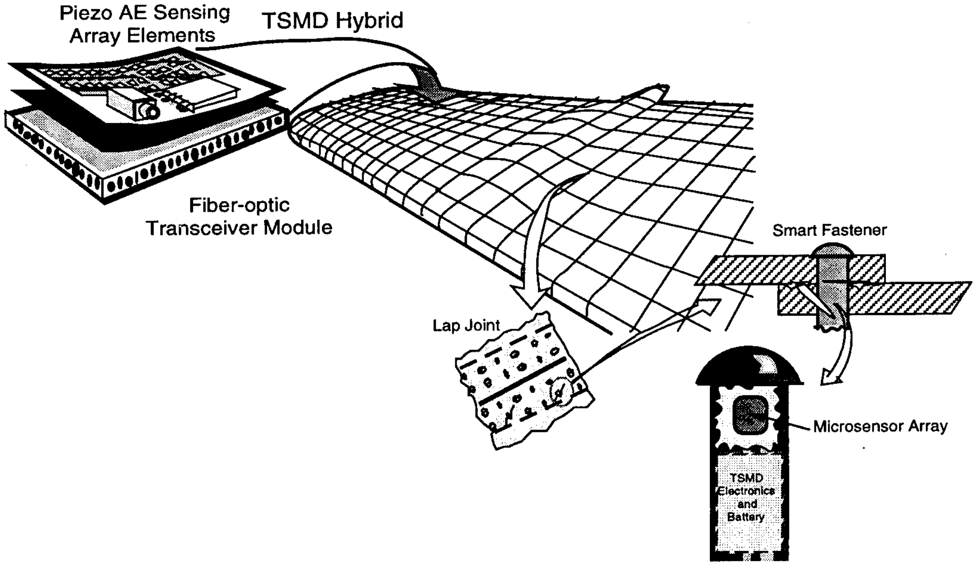

Piezo AE Sensing Array Elements
TSMD Hybrid
Fiber-optic Transceiver Module
Smart Fastener
Lap Joint
Microsensor Array
TSMD Electronics and Battery

Figure 6-8. Smart Structures Concept

6.9 VALIDATION CONSIDERATIONS FOR IVMS

Assuming sufficient benefits are anticipated to warrant combining flight-safety-critical functions with noncritical functions, this section presents some considerations regarding the validation costs that may be encountered, and offers some offsetting steps in terms of tools and techniques that may reduce these costs. In general, the cost to validate a system grows with the amount of integration. Figure 6-9 illustrates this tendency.

This relation is intuitive, representative of a movement beyond current experience. Current experience and practices stress the isolation of flight-safety-critical functions from noncritical or mission-crucial functions. However, the benefits of integration have not yet warranted going much beyond the isolation point. An example that did go beyond strict isolation was the dual-port RAM in the F-18 High Alpha Research Vehicle (HARV). It served as an interface between the flight-critical DFCS and the non-flight-critical research flight control system.

The types of new system checkout tools that will improve efficiency in validation testing, and thereby reduce costs of validation of the more integrated systems, range from automated analysis tools to automated testing and documentation. These are described in the subsequent sections.

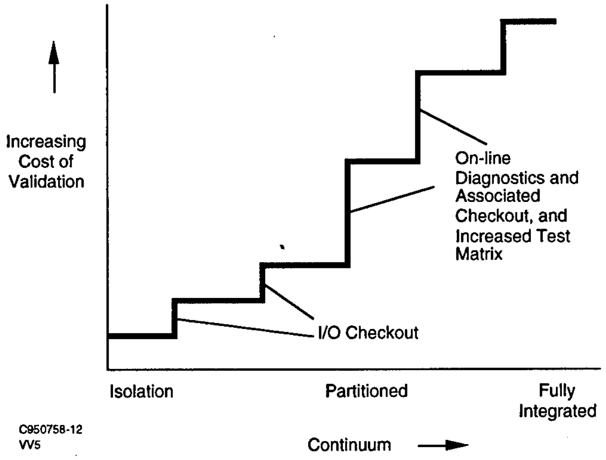


Figure 6-9. Relative System Validation Cost

6.9.1 Automated Analysis Tools

6.9.1.1 Dynamic Fault Trees

One such approach developed at UCLA is called Dynamic Flow Graph Methodology. It is an approach to modeling and analyzing software-based control systems for the purpose of reliability/safety assessment. Models representing causal and timing relationships between software functions, interfacing hardware, and external system parameters are analyzed to produce "timed" fault trees that relate the values of system variables at discrete pin points in time.

6.9.1.2 Software Evaluation Tools

Software evaluation tools include automated looping checks and complexity assessment. These tools provide measures of complexity, typing, looping, and branches.

6.9.1.3 Modification Impact Tools

Software modification impact tools identify tests that must be rerun upon a given coding modification. A model of the software can be run or can simulate the software operation. The effects of modifications and tests that need to be run will be outputs of such a tool.

6.9.2 Automated Testing and Documentation Tools

6.9.2.1 Automated Testing Tools

Examples include Computer-Aided System Tools (CAST), which provide user-friendly workstation-based environments that facilitate rapid excitation and test recording. Another example is FAST (a McDonnell Douglas system developed to automate the functional checkout of the F-18 flight control software).

6.9.2.2 Automated Documentation Tools

Although primary focus is on the design, coding, and associated documentation, these tools are also valuable for documentation and cataloging of test results. The tools also help in software design and can aid in the requirements specifications.

6.9.2.3 Automated Code Generation

Tools are currently available that will generate code from block diagrams such as ISI - MATRIX. Once the tool has been validated, the amount of testing may be able to be reduced.

6.10 SOFTWARE DESIGN METHODOLOGIES

Software development activity is actually one of the most complex human activities. Programs during the last two decades have grown from hundreds of lines of code to hundreds of thousands of lines.

The software system of an IVMS is surely a complex and large system, taking into account the number of variables, the amount of processing involved, and the level of confidence and quality the final product must achieve. To cope with those problems, tools and methodologies have been produced, reshaped, and improved, but others are now becoming available.

Phasing the development process and partitioning the software are ways of dealing with complexity. This permits groups of staff to work on the same product without increasing the number of errors. One can consider in the development phase: requirements analysis, system specifications, system design, coding, and linking.

Verification and validation during all processes are essential to improve the correctness of transitioning information from phase to phase and person to person. These and fault removal using live data are important and very expensive tasks. This permits the removal of all detected errors or infractions as soon as possible in the development process and leads to a better final documentation.

Diversity, such as N version, is also a common method used for producing high-quality software for safety-critical systems or subsystems. This has the expense for developing each version plus the expense of extra final testing to choose the selected primary version.

Formal methods of software development are in a broad sense the use of mathematical notations, models, or algebras to represent and operate information to be converted or imbedded into a software system. The use of set theory and logic to describe system specifications and software designs are the language of this method. The UK Defense Standard 00-55 has made formal methods virtually mandatory

Structural programming has already been used successfully over the years, improving the quality of software products, eliminating error sources, and improving testability. One can expect better specifications and programming, resulting in fewer errors and improving efficiency in the process and in the product. The effort spent on verification, validation, and testing can be considerably reduced, leading to cost savings.

If one can get a language more precise and universal than natural languages, it must be used as much as possible among the software systems community.

Affordability during all life cycles will be very much improved, and complex, large systems became technically and finally feasible.

6.11 DIVERSITY

Diversity is the existence of different means of performing a required function. It may be a difference in physical phenomena such as electronic and mechanical, analog and digital. It may also be a difference in specification, such as plain English, pseudo code, or formally mathematics. The difference may also be in design teams, such as independent V&V or team A and team B.

Diversity is well established in the aircraft industry as a defense against common-mode failures or design errors. Classical examples include the use of standby instruments, the provision of air-driven turbines to back up the engine-driven generators, and the mechanically driven spoilers in fly-by-wire systems.

Diversity is used in other industries, notably in nuclear power plants where some regulators require that diversity is used in the more critical systems. For example, a protection system may be required to use diverse physical phenomena to detect fault conditions, perhaps temperature and pressure.

Diversity can be achieved at various levels. It can range from completely different and separate systems to the use of different components in an otherwise common design.

Diversity has been applied in avionics systems that meet a common requirement specification with two or more different implementations that are cross monitored. The subsystems have different lower level specifications, are developed by separate and different teams, use different components, and have different software using different languages, different compilers, and different processors.

The objective of diversity is to provide protection against common mode errors. System complexity has increased because of increased integration and the application of software-based systems to provide cost-effective, increased functionality. Where such complexity is combined with safety criticality, it becomes necessary to address the issues of common mode errors in design and in implementation. Diversity is one important technique to provide part of this protection.

The Flap and Slat systems on the A310, A320, and A321 are examples of systems in which diversity has been applied at the level of different software, different languages, different development teams, and different microprocessors. Such an approach provides protection against errors in software development and in the interpretation of requirements.

In some industries, a limit is set on the level of reliability that can be claimed for a single design. The regulator will not permit a supplier to claim that a design is free from common mode errors, however thorough the design process. Examples of such regulators include UK's MOD and Nuclear Power Inspectorate. The UK's CEGB Guidelines [6.3] for reactor protection systems limit the allowable claim for a nondiverse system to 1 in 10^4 demands. UK's MOD Interim Defense Standard 00-55 [6.2] only allows a reduction in the safety integrity level of individual lanes from the multilane level if "all the components are strictly independent." This independence must be maintained throughout "specification, design, development and maintenance." The draft international IEC [6.4] standard on electronic-safety-related systems also limits the claim that can be made for a single design system. The draft IEC standard allows a reduction in safety level only if independence is shown; such independence requires that the subsystems be conceptually different, implemented using different technology, and reliant on different properties. In such a regulatory environment, diversity is often the only way forward.

Other regulators may impose demanding verification and validation activities aimed at detecting errors; for example, FAA/JAA in DO178B[6.5]. MIL-STD-882C [6.1] classifies software that controls "potentially hazardous hardware" as Control Category I unless there are independent safety systems. This leads to a Hazard Risk Index of 1 and hence to "significant analysis and testing resources." In some instances, it may be more economical and more convincing to use diversity. A particular example is the Boeing 777 PFCS, which uses three different compilers allied with three different microprocessors to provide protection against compiler errors. This approach is an alternative to providing a compiler developed to the standards of DO 1783 level A or to carrying out the verification at object code level.

Integrated vehicle management systems are and will continue to be complex, both at the functional and implementation level. The judicious use of diversity will provide protection, in safety-critical systems, against common mode errors in design and in implementation. The extra costs associated with diversity may be justified either as the only means of meeting the requirements or as alternative to very extensive verification.

6.12 CONTROLS AND DISPLAYS

The aircrew interface of a Vehicle Management System (VMS) imposes an extremely challenging set of operational and technical requirements. This interface has previously been implemented by a set of controls and displays. However, new technology has expanded the possible modalities of the aircrew-vehicle interface to include interactive tactile and audio aspects. Speech recognition technology, for instance, will be used in the near future for some functions. The applicability of these less mature technologies, particularly for flight-critical functions, is to be determined as VMS systems are configured.

The primary technical challenge in design of the control and display function of a VMS is to meet flight safety requirements. These considerations drive not only the hardware and software integrity requirements but, just as important, the manner in which the aircrew operates with them to ensure the absence of mistakes through misinterpretation of displayed data or inadvertent actuation of a control.

Implicit in the control and display design for a VMS is the need for functional integration. The possibilities for single-point failures are great in such highly integrated designs. Hence much attention must be given to designing the primary, secondary, tertiary . . . modes of operation.

Discrete as well as continuous controls for aircraft application have rapidly evolved from electromechanical switches and rheostats to touch-in, bits-out digital devices. The design flexibility of such devices is well suited to integrated systems where cockpit panel space is at a premium. The level of functional integration of a set of controls can be greatly enhanced by appropriate integration with the display unit with major emphasis on the interactive aspects of operation. In fact, the design of a control and display configuration proceeds as an integrated design.

Similarly, the era of the dedicated electromechanical display has been overtaken by highly integrated cockpit displays. The amount of information that can be reliably displayed to the aircrew with electronic displays is vast. The limiting characteristics are display surface area, resolution, color versus monochrome, and refresh rate. The functionality is significantly enhanced in a "heads-up display" configuration that allows the aircrew to view air vehicle symbolic and quantitative data in the presence of a real-world and/or enhanced out-the-windscreen view.

The well-understood operational and technical characteristics of CRT displays will serve as a baseline for specification and evaluation of succeeding generations of displays. These include active-matrix liquid crystal display (AMLCD) and active-matrix display electroluminescent (AMEL).

6.13 SUMMARY

A broad spectrum of technologies is required to configure an IVMS. As evidenced in this chapter, the technologies cover the range from hardware to software with fault tolerance and high integrity being the driving factors. Technologies that can be applied to IVMS system development evolve over time, and their applicability must be continually evaluated. The development of IVMS-specific technologies is identified by system requirements and identify where additional research is required. The IVMS technologist must continually assess new and emerging technologies to determine the applicability to a specific IVMS system application. The evaluation methodologies described in Chapter 3 can be used as guidelines in making the assessments.

6.14 REFERENCES

[6.1] MIL-STD-882C Military Standard, “System Safety Program Requirements.”

[6..2] UK MOD INT DEF STAN 00-55, “Hazard Analysis and Safety Classification of the Computer Programmable Electronic System Elements of Defense Equipment.”

[6.3] Central Electricity Generating Board, “Guidelines for the Use of Programmable Electronic Systems for Reactor Protection.”

[6.4] International ElectroTechnical Commission Technical Committee 65A Draft, “Functional Safety of Electrical/Electronic/Programmable Electronic Systems: Generic Aspects 65A (Secretariat) A23.”

[6.5] RTCA/EUROCAE DO178B/ED-12B, “Software Considerations in Airborne Systems and Equipment Certification Reliability Requirements.”

[6.6] Driscoll, K. R., Papadopoulos, G., Nelson, S., and Hartman, G., “Multi-Microprocessor Flight Control System II,” AFWAL Final Report for Contract F33615-81-C-3614.

[6.7] Airlines Electronic Engineering Committee, Supplement 10, “ARINC Specification 429: Mark 33 Digital Information Transfer System (DITS),” March 16, 1987.

[6.8] Airlines Electronic Engineering Committee, “ARINC Specification 629: Multi-Transmitter Data Bus. Part I: Technical Description,” March 7, 1990.

[6.9] “ARINC 629 Technical Overview,” *Proceedings of the 1988 Radio Technical Commission for Aeronautics Assembly*, pp. 201-211.

[6.10] IEEE Standard P1194.1, “Electrical Characteristics of Backplane Transceiver Logic,” December 1990.

[6.11] Spitzer, C., “Digital Avionics Systems,” Prentice Hall, Englewood Cliffs, NJ, 1987.

[6.12] Hoyme, K., and Driscoll, K., "SAFEbus," *Proceedings of the 11th Digital Avionics Systems Conference*, October 5-9, 1992, and *IEEE Aerospace Electronics and Systems Magazine*, March 1993.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 CONCLUSIONS

The working group made a number of conclusions concerning integrated vehicle management systems.

1. *Distinctions between VMS and Non-VMS*—Discussions concerning the distinction between vehicle management systems (VMS) and non-VMS portions of vehicle systems resulted in the definition of certain classifications:

   a. Functional distinctions as discussed in Chapter 1;

   b. Separation based upon criticality.

   It was apparent that criticality mixing posed the greatest challenge and potential for the greatest savings for integrating VMSs. The concept of mixing criticality in common module avionics became, therefore, a main theme for the working group's attention.

2. *Multilevel criticality has been the major constraint to integration of vehicle management functions*—The benefits of physical integration have been achieved for mission avionics on new vehicles such as the F-22 and RAH-66 because the level of criticality has been low. Designs of VMSs, on the other hand, have not allowed the use of common modules and other integration concepts because these were not fault tolerant.

3. *Dealing with criticality*—Numerous techniques have been used to deal with flight-critical systems:

   - N-version diversity—the concept of independent design of N channels of flight-critical functions conceivably covers failure modes. True independence, however, is extremely expensive to design and maintain. Furthermore, independence for such fault coverage has never been achieved.
   - Formal methods—A technique that promises to prove that a design has no unknown failure paths; the practice has been limited to a few very simple problems. If the concept can be matured, it could be a powerful tool to design safe, low-cost systems.
   - Robust partitioning— A concept to guarantee that no unwanted interactions occur between functions; this is difficult to do with shared physical resources such as modular avionics. New examples like the Integrated Modular Avionics on the Boeing 777 have emerged however.

4. *Design and verification guidelines for flight-critical systems are not formal.*

5. *System partitioning evolves*—Although robust partitioning and eventually formal methods will enable designs of completely integrated multilevel critical functions, the notion of one common architecture does not make sense. It is more sensible to segregate functions that have extensive serial computations and high interaction, such as flight control, navigation, stores management, and so on from high-bandwidth parallel processing functions, such as EW image processing and display drivers. This type of processing type segregation would provide even more economic incentives than a single common module architecture.

6. *Cost benefits and penalties of IVMSs*—Since the early 1980s, the cost benefits of using common modules in an extensible backplane cabinet have been projected. Reality has witnessed the measurable increases in cost of aviation electronics, however, it is believed that common modules have resulted in net lower costs. These cost benefits result from:

   - Large quantities of common parts result in economies of scale when compared to numerous separate subsystem designs using a conventional black box design approach.
   - Commonality of designs and hardware across platforms is also quite viable.
   - The larger set of common modules enhances the likelihood that critical parts will available throughout the system life cycle.

7.2 RECOMMENDATIONS

As with most working groups, more questions arise during the course of deliberations than answers. Discussion concerning IVMSs yielded many future challenges. Three form of recommendations are made. Further research and development is needed in certain areas to achieve significant payoffs for IVMS. Second, additional working group activities that lead to standards are advised.

Research and Development—Formal methods have received much attention in the past two decades. Because a true formal method would eliminate the desire to solve flight-critical problems with N-version techniques, addition R&D would be justified. A second area of research is standards. Standards for interfaces, networks, processing, and software performed by designers would lead to clearer and more useful design guidelines.

Working Groups—New working group efforts would also benefit the NATO technical community. Working group guidelines in robust partitioning, verification and validation, and system partitioning for integrated systems would have a big payoff.

APPENDIX A

MATERIAL PRESENTED TO THE WORKING GROUP DURING MEETINGS

A.1 GUEST SPEAKERS

The following were guest speakers invited to the WG meetings; included for each are the organization and the topic presented:

Dr. Chun Lam, Allied Signal Canada, Integrated Closed Environmental Control System and Utility Subsystem Integration

Dr. Jay Lala, Draper Lab, Advanced Information Processing System

Mr. John Ostgaard, Wright Laboratory, Pave Pillar/Pave Pace

Mr. Bill Leonard, Lockheed Ft. Worth, YF-22 and F-22 VMS

James McWha, Boeing Commercial, 777 Flight Systems

Mr. Dagfinn Gangsaas, Boeing, VMS Systems

A.2 PRESENTATIONS BY WG MEMBERS

The following represents briefings given by members of the WG:

E. Stear, Formal Methods

K. Driscoll, Boeing 777 SAFEbus

G. Belcher, Boeing 777 Primary Flight Computers

G. Belcher, N-Version Design

M. Stieglitz, Baseline Software Approaches

J. Kaul, European Fighter Aircraft Control System

D. Deets, Error Reporting During Design and Test

A. DeThomas, VMS Background and SAE Activities

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's Reference	3. Further Reference	4. Security Classification of Document
	AGARD-AR-343	ISBN 92-836-1035-0	UNCLASSIFIED/ UNLIMITED

5. Originator Advisory Group for Aerospace Research and Development
North Atlantic Treaty Organization
7 rue Ancelle, 92200 Neuilly-sur-Seine, France

6. Title

Integrated Vehicle Management Systems

7. Presented at/sponsored by

8. Author(s)/Editor(s)	9. Date
	April 1996

10. Author's/Editor's Address	11. Pages
	140

12. Distribution Statement There are no restrictions on the distribution of this document. Information about the availability of this and other AGARD unclassified publications is given on the back cover.

13. Keywords/Descriptors

Technology
Weapon systems
Integrated systems
Electronics systems
Design
Fault tolerance
Safety
Aerospace engineering
Flight control
Navigation
Propulsion
Aircraft
Helicopters
Missiles
UAV (Unmanned Aerial Vehicle)

14. Abstract

Major trends in technology, weapon system performance goals and affordability for aerospace systems are occurring simultaneously. For avionic systems this performance and affordability can be achieved by functional and physical integration. "Functionally" integrated subsystems to achieve higher performance has been greatly aided by advances in computer technology. The desire to minimize costs for these systems has been accomplished through a "physical" integration concept based upon common modules tied through a high speed backplane. The concept, called integrated avionics, has been used on new aircraft such as the US Air Force F-22 fighter and the Boeing 777 commercial transport.

Vehicle management systems provide the management of crucial flight functions and systems for advanced aerospace vehicles. These systems must have high integrity, safety, and overall fault tolerance. Low cost modular avionics are unproven for such fault tolerant systems. This becomes a key issue for investigation.

This report deals with the key problems in fault tolerance for modular computer based systems. New techniques, only recently applied, provide exciting possibilities to reduce avionics costs and maintain high integrity and safety. These techniques and more are discussed in this report sponsored by the Mission Systems Panel of the AGARD.

AGARD

NATO - OTAN

7 RUE ANCELLE • 92200 NEUILLY-SUR-SEINE

FRANCE

Télécopie (1)47.38.57.99 • Télex 610 176

DIFFUSION DES PUBLICATIONS

AGARD NON CLASSIFIEES

Aucun stock de publications n'a existé à AGARD. A partir de 1993, AGARD détiendra un stock limité des publications associées aux cycles de conférences et cours spéciaux ainsi que les AGARDographies et les rapports des groupes de travail, organisés et publiés à partir de 1993 inclus. Les demandes de renseignements doivent être adressées à AGARD par lettre ou par fax à l'adresse indiquée ci-dessus. *Veuillez ne pas téléphoner*. La diffusion initiale de toutes les publications de l'AGARD est effectuée auprès des pays membres de l'OTAN par l'intermédiaire des centres de distribution nationaux indiqués ci-dessous. Des exemplaires supplémentaires peuvent parfois être obtenus auprès de ces centres (à l'exception des Etats-Unis). Si vous souhaitez recevoir toutes les publications de l'AGARD, ou simplement celles qui concernent certains Panels, vous pouvez demander à être inclu sur la liste d'envoi de l'un de ces centres. Les publications de l'AGARD sont en vente auprès des agences indiquées ci-dessous, sous forme de photocopie ou de microfiche.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Fachinformationszentrum Karlsruhe
D-76344 Eggenstein-Leopoldshafen 2

BELGIQUE
Coordonnateur AGARD-VSL
Etat-major de la Force aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

CANADA
Directeur, Services d'information scientifique
Ministère de la Défense nationale
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Defence Research Establishment
Ryvangs Allé 1
P.O. Box 2715
DK-2100 Copenhagen Ø

ESPAGNE
INTA (AGARD Publications)
Pintor Rosales 34
28008 Madrid

ETATS-UNIS
NASA Headquarters
Code JOB-1
Washington, D.C. 20546

FRANCE
O.N.E.R.A. (Direction)
29, Avenue de la Division Leclerc
92322 Châtillon Cedex

GRECE
Hellenic Air Force
Air War College
Scientific and Technical Library
Dekelia Air Force Base
Dekelia, Athens TGA 1010

ISLANDE
Director of Aviation
c/o Flugrad
Reykjavik

ITALIE
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
Aeroporto Pratica di Mare
00040 Pomezia (Roma)

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
N-2007 Kjeller

PAYS-BAS
Netherlands Delegation to AGARD
National Aerospace Laboratory NLR
P.O. Box 90502
1006 BM Amsterdam

PORTUGAL
Estado Maior da Força Aérea
SDFA - Centro de Documentação
Alfragide
2700 Amadora

ROYAUME-UNI
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

TURQUIE
Millî Savunma Başkanliği (MSB)
ARGE Dairesi Başkanliği (MSB)
06650 Bakanliklar-Ankara

Le centre de distribution national des Etats-Unis ne détient PAS de stocks des publications de l'AGARD.
D'éventuelles demandes de photocopies doivent être formulées directement auprès du NASA Center for AeroSpace Information (CASI) à l'adresse ci-dessous. Toute notification de changement d'adresse doit être fait également auprès de CASI.

AGENCES DE VENTE

NASA Center for
AeroSpace Information (CASI)
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
Etats-Unis

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris
France

The British Library
Document Supply Division
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
Royaume-Uni

Les demandes de microfiches ou de photocopies de documents AGARD (y compris les demandes faites auprès du CASI) doivent comporter la dénomination AGARD, ainsi que le numéro de série d'AGARD (par exemple AGARD-AG-315). Des informations analogues, telles que le titre et la date de publication sont souhaitables. Veuiller noter qu'il y a lieu de spécifier AGARD-R-nnn et AGARD-AR-nnn lors de la commande des rapports AGARD et des rapports consultatifs AGARD respectivement. Des références bibliographiques complètes ainsi que des résumés des publications AGARD figurent dans les journaux suivants:

Scientific and Technical Aerospace Reports (STAR)
publié par la NASA Scientific and Technical
Information Division
NASA Headquarters (JTT)
Washington D.C. 20546
Etats-Unis

Government Reports Announcements and Index (GRA&I)
publié par le National Technical Information Service
Springfield
Virginia 22161
Etats-Unis
(accessible également en mode interactif dans la base de
données bibliographiques en ligne du NTIS, et sur CD-ROM)

Imprimé par le Groupe Communication Canada
45, boul. Sacré-Cœur, Hull (Québec), Canada K1A 0S7

AGARD

NATO OTAN

7 RUE ANCELLE • 92200 NEUILLY-SUR-SEINE

FRANCE

Telefax (1)47.38.57.99 • Telex 610 176

DISTRIBUTION OF UNCLASSIFIED AGARD PUBLICATIONS

AGARD holds limited quantities of the publications that accompanied Lecture Series and Special Courses held in 1993 or later, and of AGARDographs and Working Group reports published from 1993 onward. For details, write or send a telefax to the address given above. *Please do not telephone.*

AGARD does not hold stocks of publications that accompanied earlier Lecture Series or Courses or of any other publications. Initial distribution of all AGARD publications is made to NATO nations through the National Distribution Centres listed below. Further copies are sometimes available from these centres (except in the United States). If you have a need to receive all AGARD publications, or just those relating to one or more specific AGARD Panels, they may be willing to include you (or your organisation) on their distribution list. AGARD publications may be purchased from the Sales Agencies listed below, in photocopy or microfiche form.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Coordonnateur AGARD — VSL
Etat-major de la Force aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

CANADA
Director Scientific Information Services
Dept of National Defence
Ottawa, Ontario K1A 0K2

DENMARK
Danish Defence Research Establishment
Ryvangs Allé 1
P.O. Box 2715
DK-2100 Copenhagen Ø

FRANCE
O.N.E.R.A. (Direction)
29 Avenue de la Division Leclerc
92322 Châtillon Cedex

GERMANY
Fachinformationszentrum Karlsruhe
D-76344 Eggenstein-Leopoldshafen 2

GREECE
Hellenic Air Force
Air War College
Scientific and Technical Library
Dekelia Air Force Base
Dekelia, Athens TGA 1010

ICELAND
Director of Aviation
c/o Flugrad
Reykjavik

ITALY
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
Aeroporto Pratica di Mare
00040 Pomezia (Roma)

LUXEMBOURG
See Belgium

NETHERLANDS
Netherlands Delegation to AGARD
National Aerospace Laboratory, NLR
P.O. Box 90502
1006 BM Amsterdam

NORWAY
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
N-2007 Kjeller

PORTUGAL
Estado Maior da Força Aérea
SDFA - Centro de Documentação
Alfragide
2700 Amadora

SPAIN
INTA (AGARD Publications)
Pintor Rosales 34
28008 Madrid

TURKEY
Millî Savunma Başkanliği (MSB)
ARGE Dairesi Başkanliği (MSB)
06650 Bakanliklar-Ankara

UNITED KINGDOM
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

UNITED STATES
NASA Headquarters
Code JOB-1
Washington, D.C. 20546

The United States National Distribution Centre does NOT hold stocks of AGARD publications.

Applications for copies should be made direct to the NASA Center for AeroSpace Information (CASI) at the address below.
Change of address requests should also go to CASI.

SALES AGENCIES

NASA Center for
AeroSpace Information (CASI)
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
United States

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris
France

The British Library
Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
United Kingdom

Requests for microfiches or photocopies of AGARD documents (including requests to CASI) should include the word 'AGARD' and the AGARD serial number (for example AGARD-AG-315). Collateral information such as title and publication date is desirable. Note that AGARD Reports and Advisory Reports should be specified as AGARD-R-nnn and AGARD-AR-nnn, respectively. Full bibliographical references and abstracts of AGARD publications are given in the following journals:

Scientific and Technical Aerospace Reports (STAR)
published by NASA Scientific and Technical
Information Division
NASA Headquarters (JTT)
Washington D.C. 20546
United States

Government Reports Announcements and Index (GRA&I)
published by the National Technical Information Service
Springfield
Virginia 22161
United States
(also available online in the NTIS Bibliographic
Database or on CD-ROM)

Printed by Canada Communication Group
45 Sacré-Cœur Blvd., Hull (Québec), Canada K1A 0S7

ISBN 92-836-1035-0